Serverless as a Bridge Between HPC and Clouds

Marcin Copik, Alexandru Calotoiu, Torsten Hoefler

IPDPS
2023 • St. Petersburg, Florida USA

Serverless on servers.

Serverless on servers.

“But serverless is slow and expensive”

"But serverless is slow and expensive"

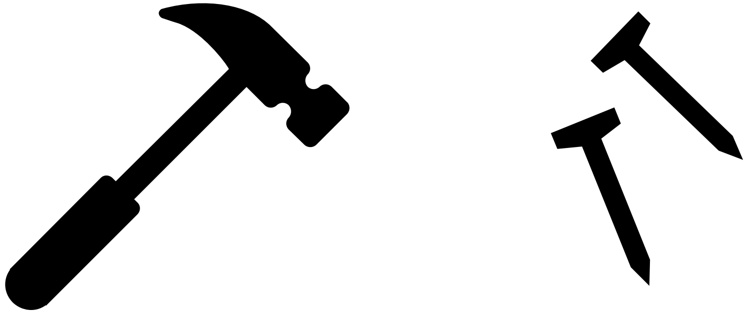
Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%

The move from a distributed microservices architecture to a monolith application helped achieve higher scale, resilience, and reduce costs.

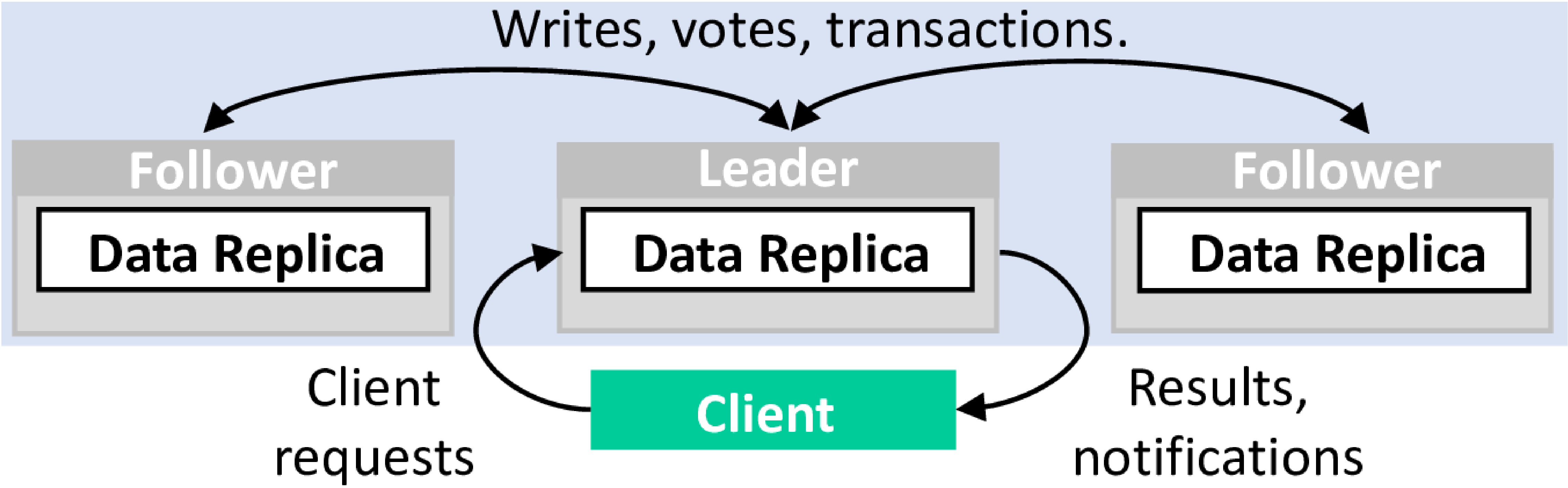
“But serverless is slow and expensive”

Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%

The move from a distributed microservices architecture to a monolith application helped achieve higher scale, resilience, and reduce costs.

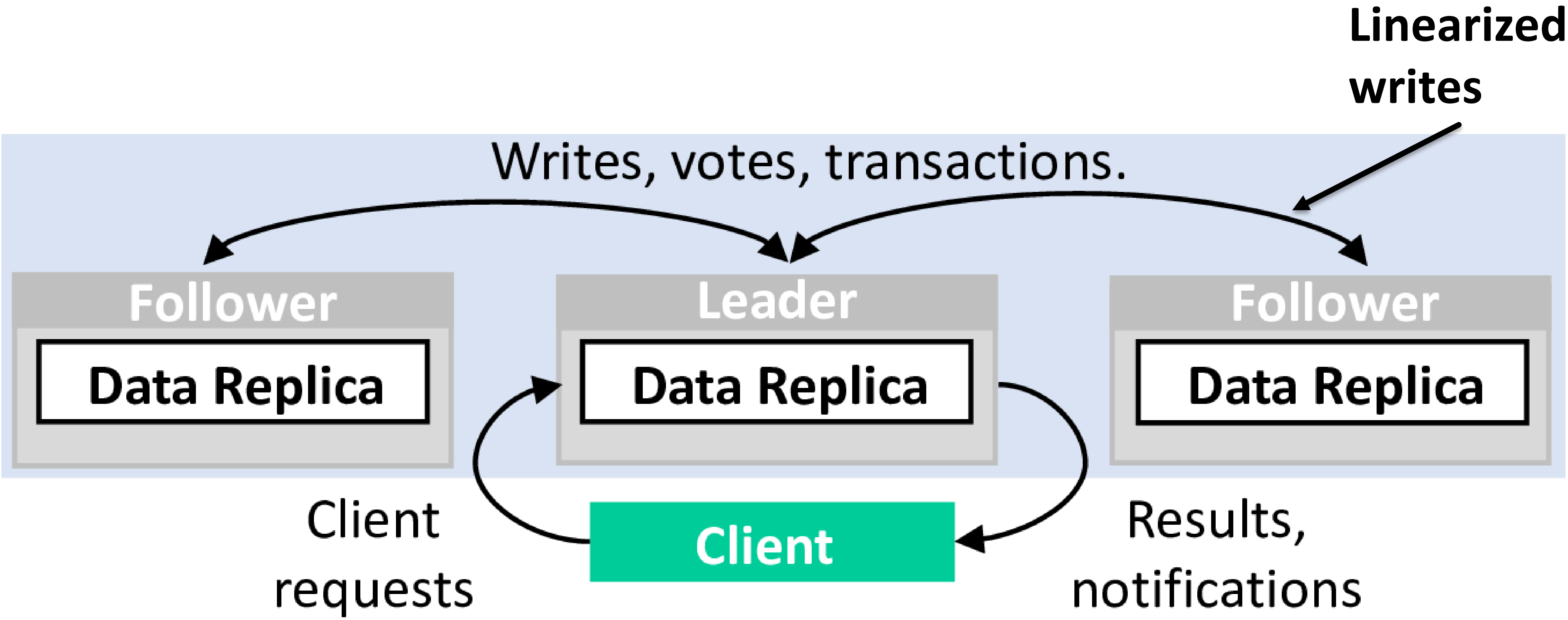


Building Serverless Services: FaaSKeeper



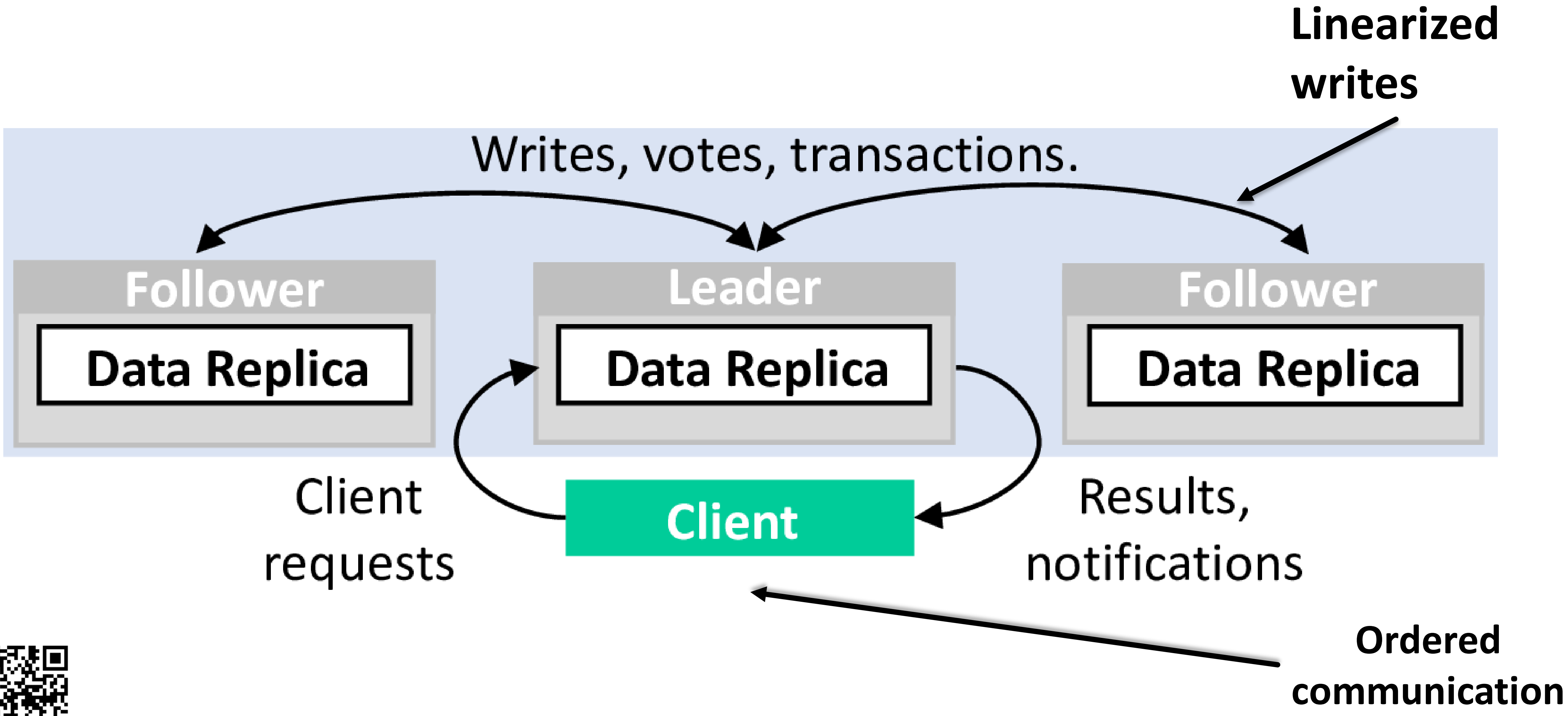
"FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example"

Building Serverless Services: FaaSKeeper



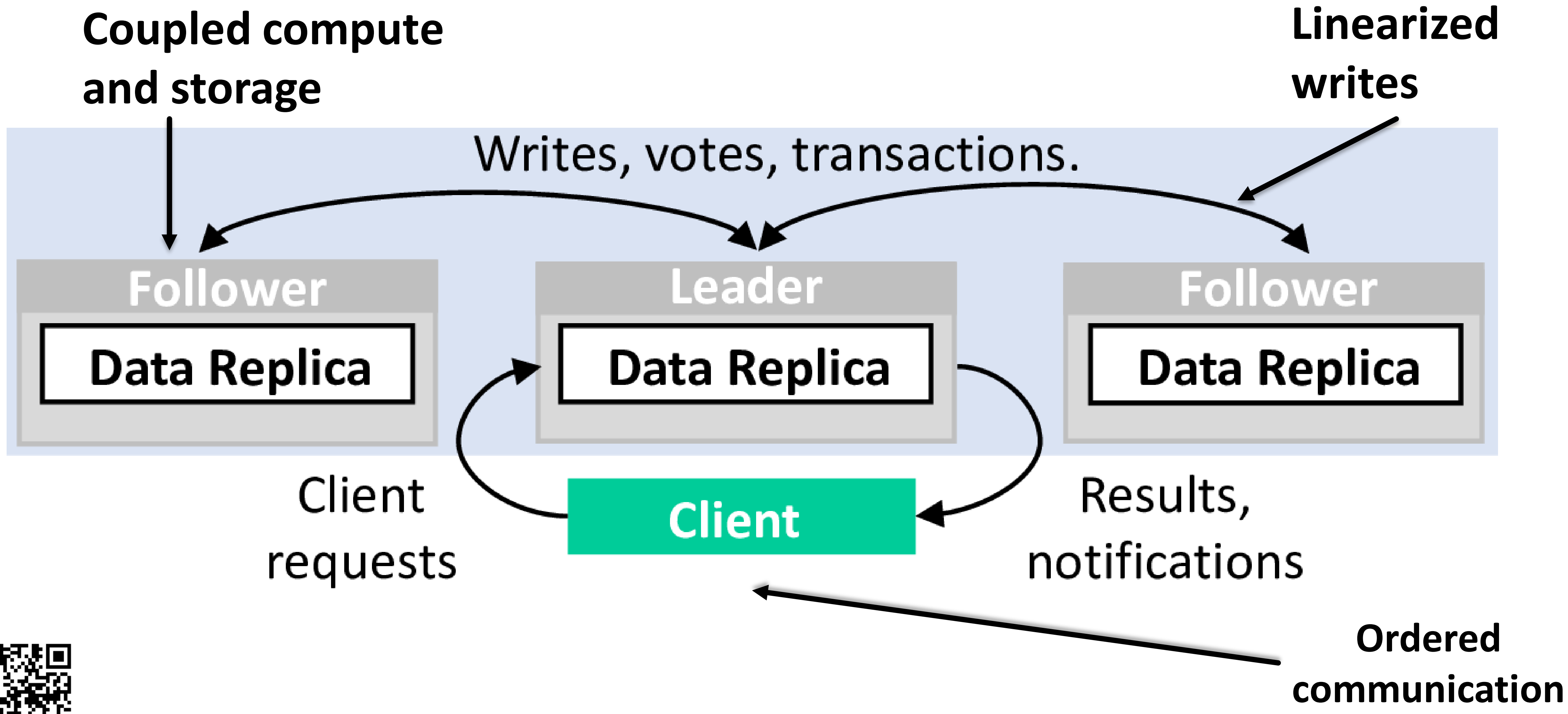
"FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example"

Building Serverless Services: FaaSKeeper



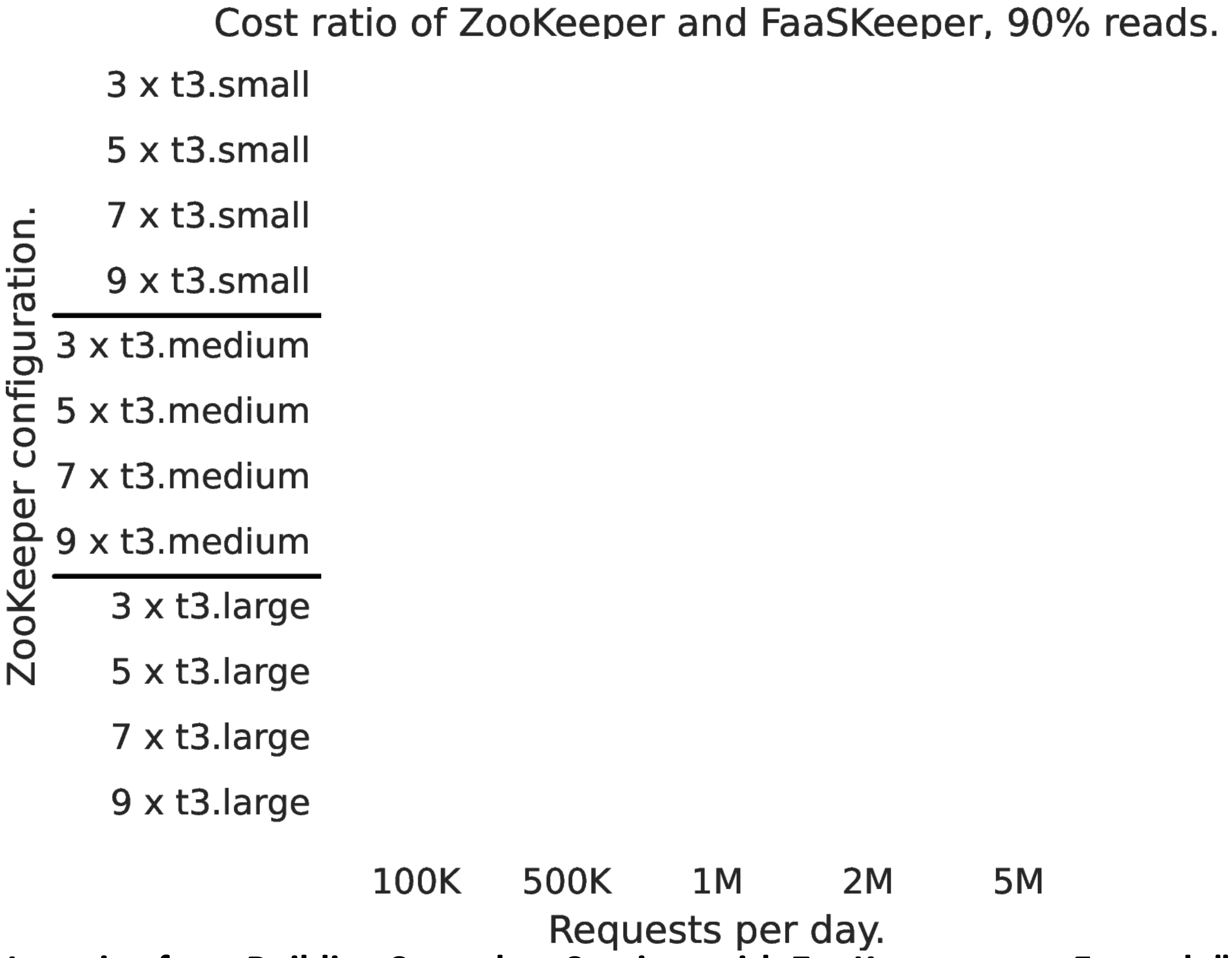
"FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example"

Building Serverless Services: FaaSKeeper



"FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example"

Building Serverless Services: FaaSKeeper



“FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example”

Building Serverless Services: FaaSKeeper

Cost ratio of ZooKeeper and FaaSKeeper, 90% reads.

ZooKeeper configuration. \ Requests per day.	100K	500K	1M	2M	5M
3 x t3.small	10.15	2.03	1.01	0.51	0.20
5 x t3.small	16.91	3.38	1.69	0.85	0.34
7 x t3.small	23.67	4.73	2.37	1.18	0.47
9 x t3.small	30.44	6.09	3.04	1.52	0.61
3 x t3.medium	20.29	4.06	2.03	1.01	0.41
5 x t3.medium	33.82	6.76	3.38	1.69	0.68
7 x t3.medium	47.35	9.47	4.73	2.37	0.95
9 x t3.medium	60.88	12.18	6.09	3.04	1.22
3 x t3.large	40.58	8.12	4.06	2.03	0.81
5 x t3.large	67.64	13.53	6.76	3.38	1.35
7 x t3.large	94.70	18.94	9.47	4.73	1.89
9 x t3.large	121.75	24.35	12.18	6.09	2.44

“FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example”

Building Serverless Services: FaaSKeeper

Cost ratio of ZooKeeper and FaaSKeeper, 90% reads.

ZooKeeper configuration. \ Requests per day.	100K	500K	1M	2M	5M
3 x t3.small	10.15	2.03	1.01	0.51	0.20
5 x t3.small	16.91	3.38	1.69	0.85	0.34
7 x t3.small	23.67	4.73	2.37	1.18	0.47
9 x t3.small	30.44	6.09	3.04	1.52	0.61
3 x t3.medium	20.29	4.06	2.03	1.01	0.41
5 x t3.medium	33.82	6.76	3.38	1.69	0.68
7 x t3.medium	47.35	9.47	4.73	2.37	0.95
9 x t3.medium	60.88	12.18	6.09	3.04	1.22
3 x t3.large	40.58	8.12	4.06	2.03	0.81
5 x t3.large	67.64	13.53	6.76	3.38	1.35
7 x t3.large	94.70	18.94	9.47	4.73	1.89
9 x t3.large	121.75	24.35	12.18	6.09	2.44

Color scale: 120.0, 60.0, 1.0, 0.5, 0.2

"FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example"

Building Serverless Services: FaaSKeeper

Cost ratio of ZooKeeper and FaaSKeeper, 90% reads.

ZooKeeper configuration. / Requests per day.	100K	500K	1M	2M	5M
3 x t3.small	10.15	2.03	1.01	0.51	0.20
5 x t3.small	16.91	3.38	1.69	0.85	0.34
7 x t3.small	23.67	4.73	2.37	1.18	0.47
9 x t3.small	30.44	6.09	3.04	1.52	0.61
3 x t3.medium	20.29	4.06	2.03	1.01	0.41
5 x t3.medium	33.82	6.76	3.38	1.69	0.68
7 x t3.medium	47.35	9.47	4.73	2.37	0.95
9 x t3.medium	60.88	12.18	6.09	3.04	1.22
3 x t3.large	40.58	8.12	4.06	2.03	0.81
5 x t3.large	67.64	13.53	6.76	3.38	1.35
7 x t3.large	94.70	18.94	9.47	4.73	1.89
9 x t3.large	121.75	24.35	12.18	6.09	2.44

Color scale: 120.0, 60.0, 1.0, 0.5, 0.2

"FaaSKeeper: Learning from Building Serverless Services with ZooKeeper as an Example"

Tracking Wasted Money in HPC

Tracking Wasted Money in HPC

FINAL REPORT
WORKLOAD ANALYSIS OF BLUE WATERS
(ACI 1650758)

Matthew D. Jones, Joseph P. White, Martins Innus, Robert L. DeLeon, Nikolay Simakov, Jeffrey T. Palmer, Steven M. Gallo, and Thomas R. Furlani (furlani@buffalo.edu), Center for Computational Research, University at Buffalo, SUNY

Michael Showerman, Robert Brunner, Andry Kot, Gregory Bauer, Brett Bode, Jeremy Enos, and William Kramer (wtkramer@illinois.edu), National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana Champaign

arXiv, 2017

Job Characteristics on Large-Scale Systems: Long-Term Analysis, Quantification, and Implications*

Tirthak Patel
Northeastern University

Zhengchun Liu, Raj Kettimuthu
Argonne National Laboratory

Paul Rich, William Allcock
Argonne National Laboratory

Devesh Tiwari
Northeastern University

SC, 2020

Comprehensive Workload Analysis and Modeling of a Petascale Supercomputer

Haihang You[1] and Hao Zhang[2]

[1] National Institute for Computational Sciences,
Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[2] Department of Electrical Engineering and Computer Science,
University of Tennessee, Knoxville, TN 37996, USA
{hyou,haozhang}@utk.edu

JSSPP, 2012

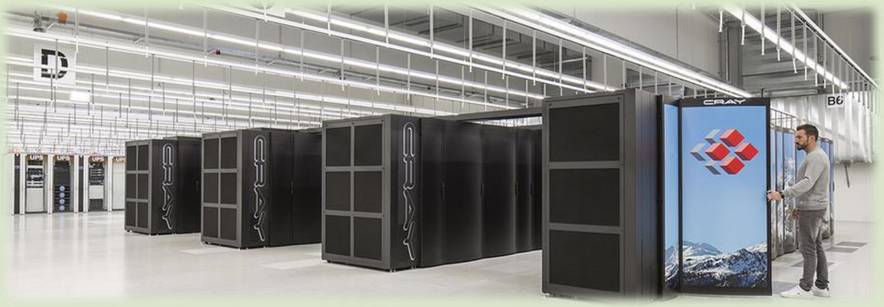
HPC System Utilization - CPU

HPC System Utilization - CPU

Piz Daint, April 2022.
- XC50 nodes – CPU + GPU, 64 GB memory.
- XC40 nodes – CPU, 64/128 GB memory.

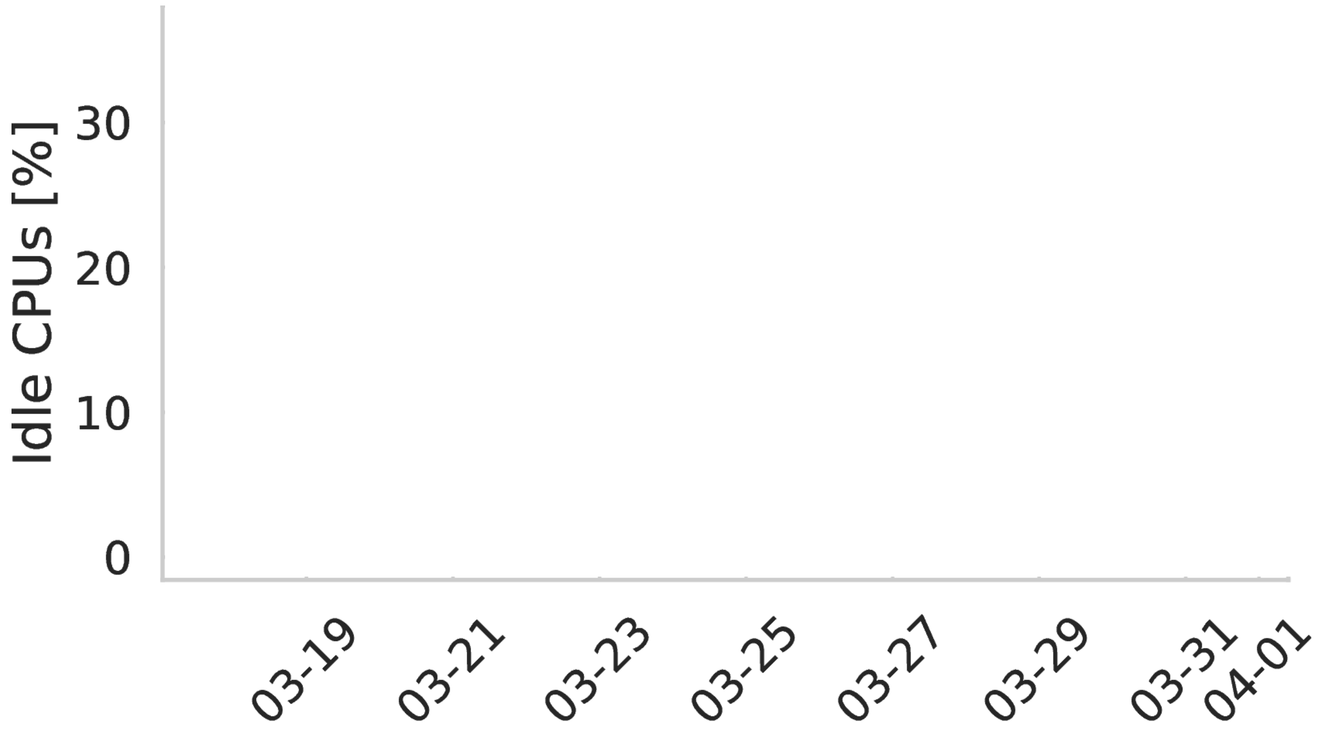
Query SLURM info every two minutes.

HPC System Utilization - CPU

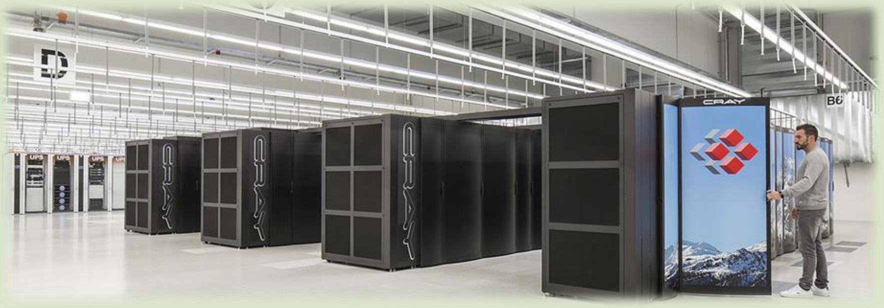


Piz Daint, April 2022.
- XC50 nodes – CPU + GPU, 64 GB memory.
- XC40 nodes – CPU, 64/128 GB memory.

Query SLURM info every two minutes.



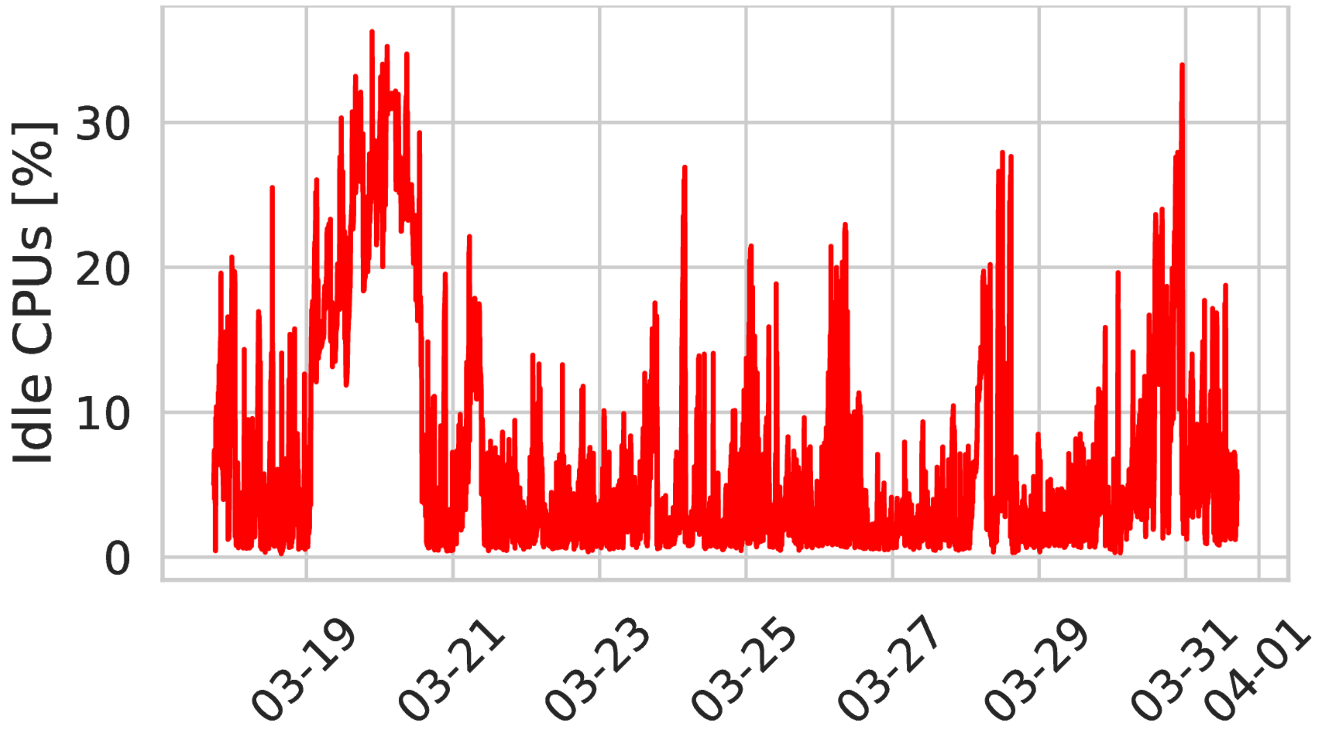
HPC System Utilization - CPU



Piz Daint, April 2022.

- XC50 nodes – CPU + GPU, 64 GB memory.
- XC40 nodes – CPU, 64/128 GB memory.

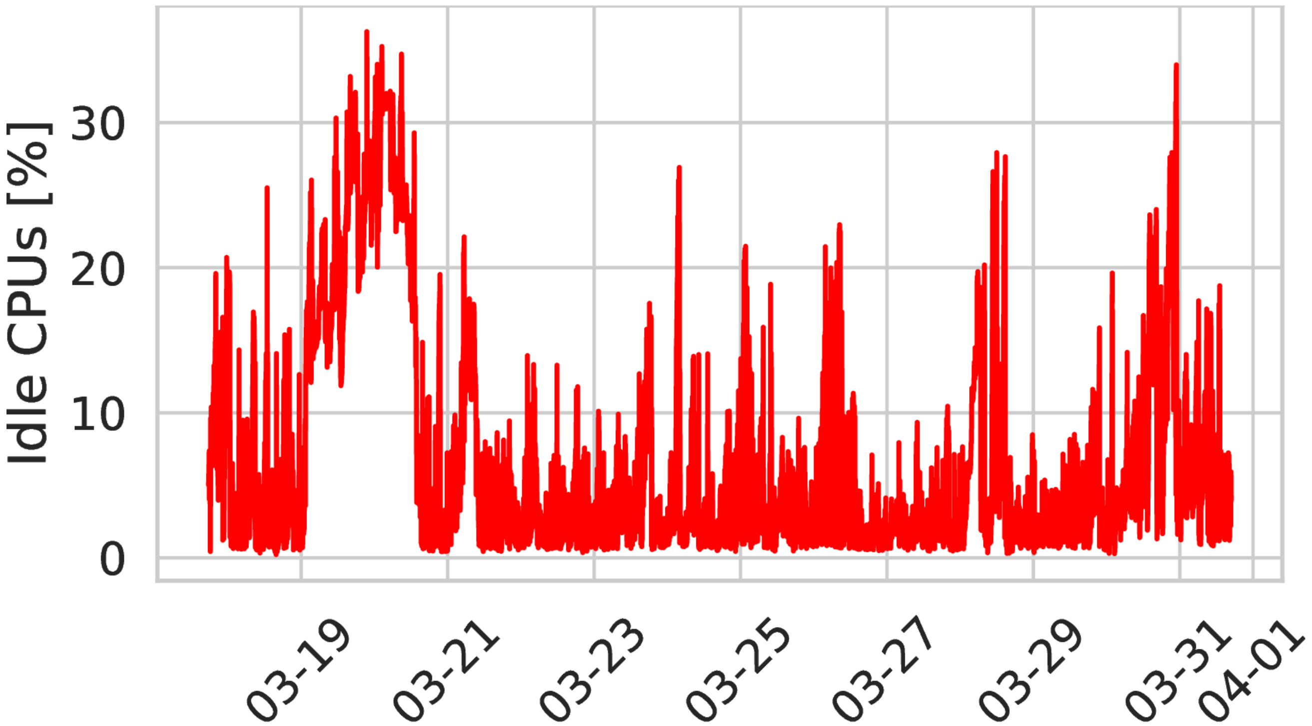
Query SLURM info every two minutes.



HPC System Utilization - CPU

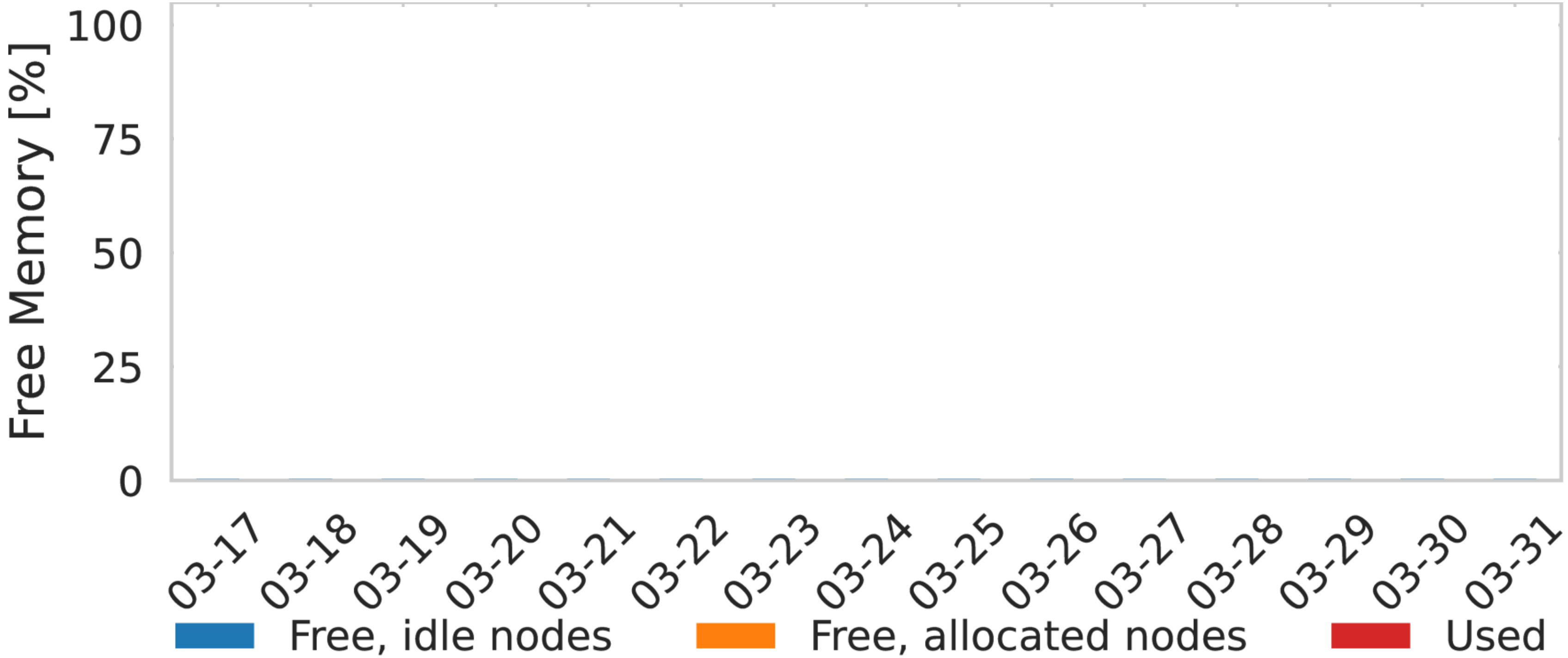
Piz Daint, April 2022.
- XC50 nodes – CPU + GPU, 64 GB memory.
- XC40 nodes – CPU, 64/128 GB memory.

Query SLURM info every two minutes.

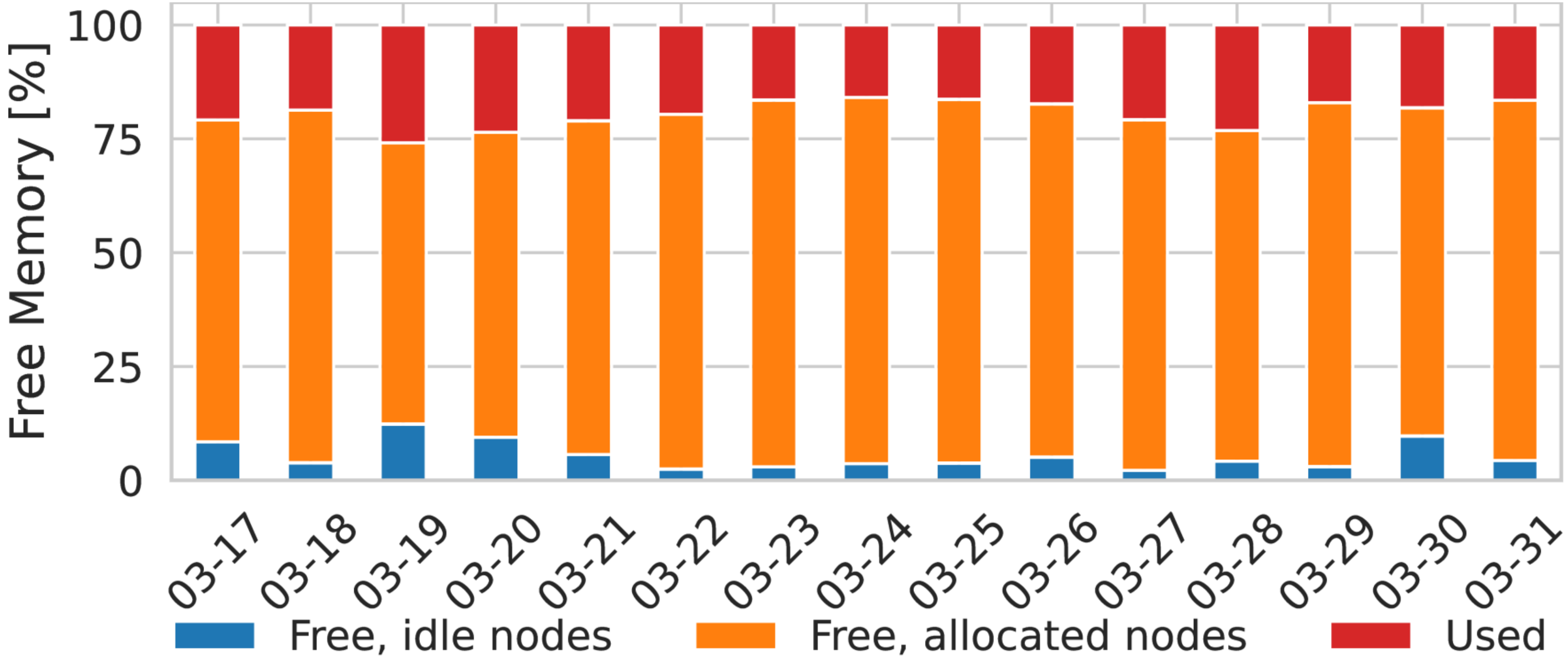


Nodes do not stay idle for long.

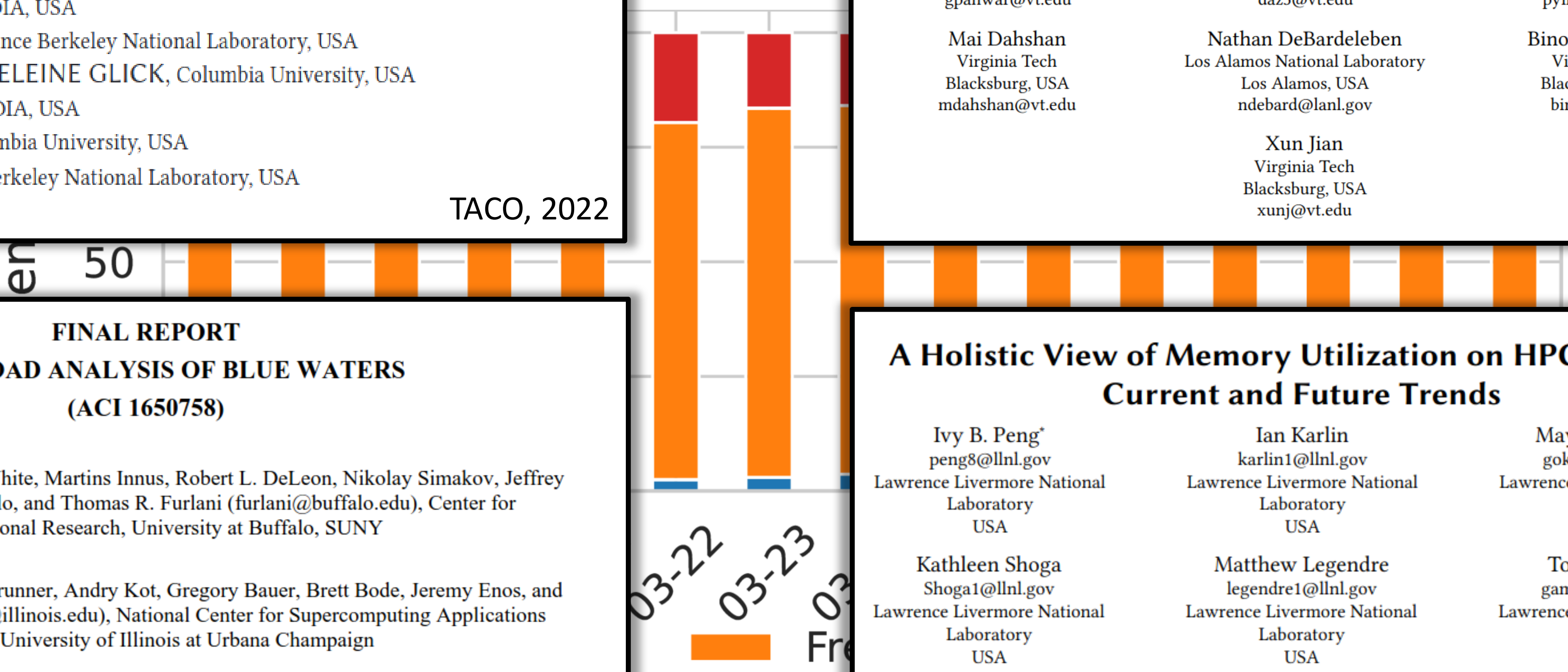
HPC System Utilization - Memory



HPC System Utilization - Memory



HPC System Utilization - Memory



A Case For Intra-rack Resource Disaggregation in HPC

GEORGE MICHELOGIANNAKIS, Lawrence Berkeley National Laboratory, USA
BENJAMIN KLENK, NVIDIA, USA
BRANDON COOK, Lawrence Berkeley National Laboratory, USA
MIN YEE TEH and MADELEINE GLICK, Columbia University, USA
LARRY DENNISON, NVIDIA, USA
KEREN BERGMAN, Columbia University, USA
JOHN SHALF, Lawrence Berkeley National Laboratory, USA

TACO, 2022

Quantifying Memory Underutilization in HPC Systems and Using it to Improve Performance via Architecture Support

Gagandeep Panwar*, Virginia Tech, Blacksburg, USA, gpanwar@vt.edu
Da Zhang*, Virginia Tech, Blacksburg, USA, daz3@vt.edu
Yihan Pang*, Virginia Tech, Blacksburg, USA, pyihan1@vt.edu
Mai Dahshan, Virginia Tech, Blacksburg, USA, mdahshan@vt.edu
Nathan DeBardeleben, Los Alamos National Laboratory, Los Alamos, USA, ndebard@lanl.gov
Binoy Ravindran, Virginia Tech, Blacksburg, USA, binoy@vt.edu
Xun Jian, Virginia Tech, Blacksburg, USA, xunj@vt.edu

MICRO, 2019

FINAL REPORT
WORKLOAD ANALYSIS OF BLUE WATERS
(ACI 1650758)

Matthew D. Jones, Joseph P. White, Martins Innus, Robert L. DeLeon, Nikolay Simakov, Jeffrey T. Palmer, Steven M. Gallo, and Thomas R. Furlani (furlani@buffalo.edu), Center for Computational Research, University at Buffalo, SUNY

Michael Showerman, Robert Brunner, Andry Kot, Gregory Bauer, Brett Bode, Jeremy Enos, and William Kramer (wtkramer@illinois.edu), National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana Champaign

arXiv, 2017

A Holistic View of Memory Utilization on HPC Systems: Current and Future Trends

Ivy B. Peng*, peng8@llnl.gov, Lawrence Livermore National Laboratory, USA
Ian Karlin, karlin1@llnl.gov, Lawrence Livermore National Laboratory, USA
Maya B. Gokhale, gokhale2@llnl.gov, Lawrence Livermore National Laboratory, USA
Kathleen Shoga, Shoga1@llnl.gov, Lawrence Livermore National Laboratory, USA
Matthew Legendre, legendre1@llnl.gov, Lawrence Livermore National Laboratory, USA
Todd Gamblin, gamblin2@llnl.gov, Lawrence Livermore National Laboratory, USA

MEMSYS, 2021

FaaS in High-Performance Applications

FaaS in High-Performance Applications

Serverless is slow

FaaS in High-Performance Applications

Serverless is slow

Communication is slow and restricted

FaaS in High-Performance Applications

Serverless is slow

Communication is slow and restricted

Serverless is hard to program.

FaaS in High-Performance Applications

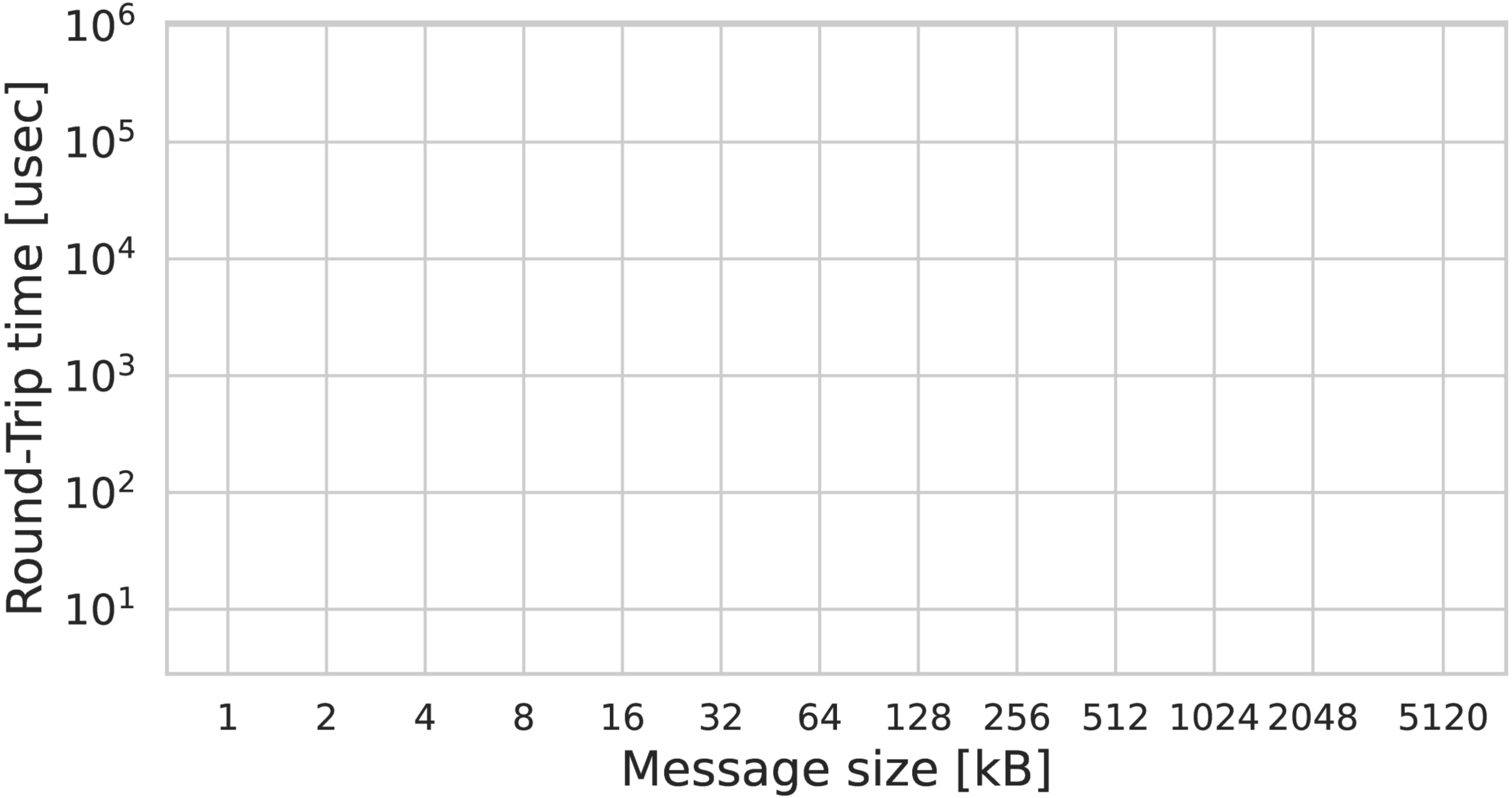
Serverless is slow

Communication is slow and restricted

Answer: rFaaS

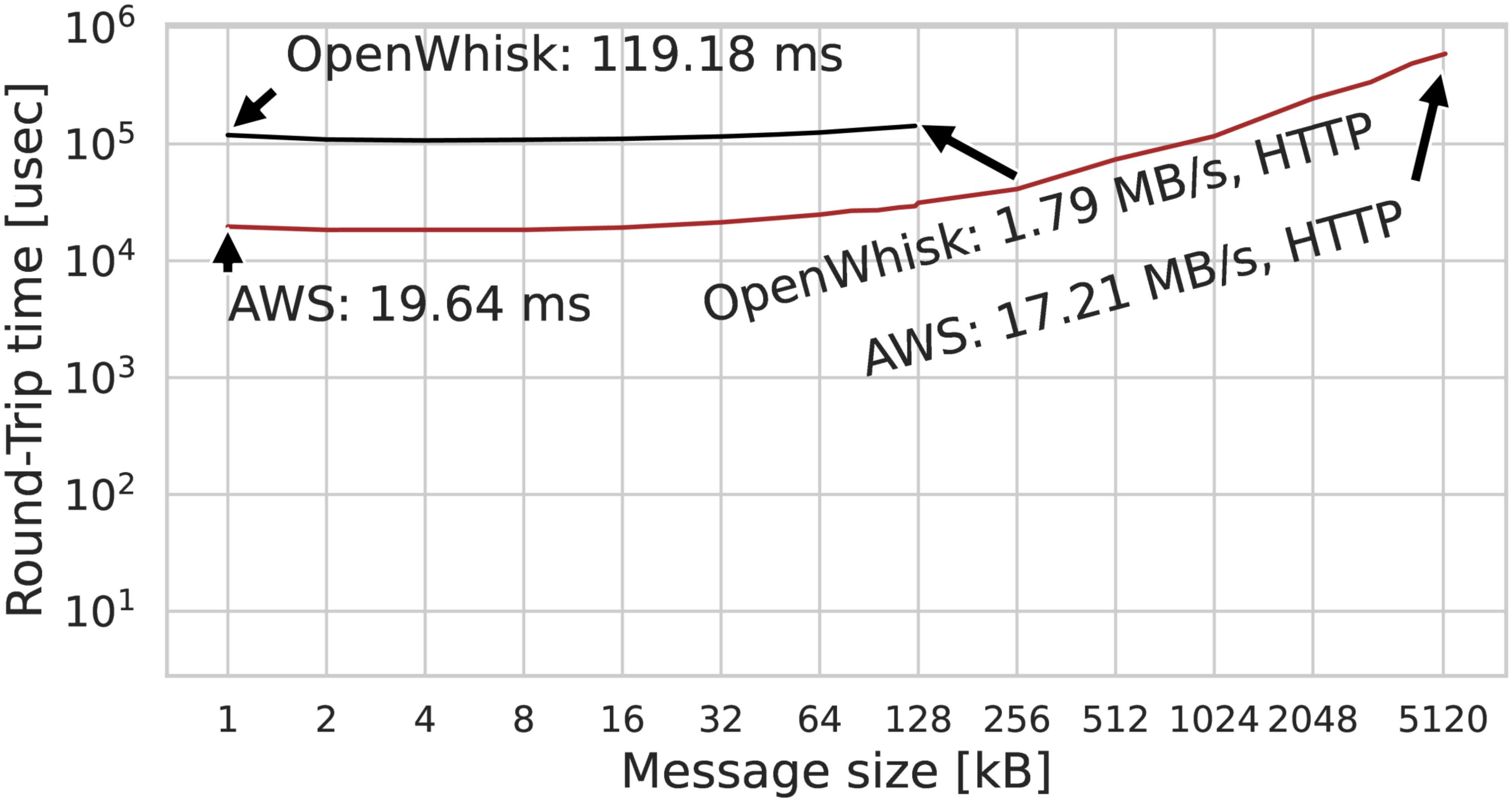
Serverless is hard to program.

How fast are invocations in FaaS?



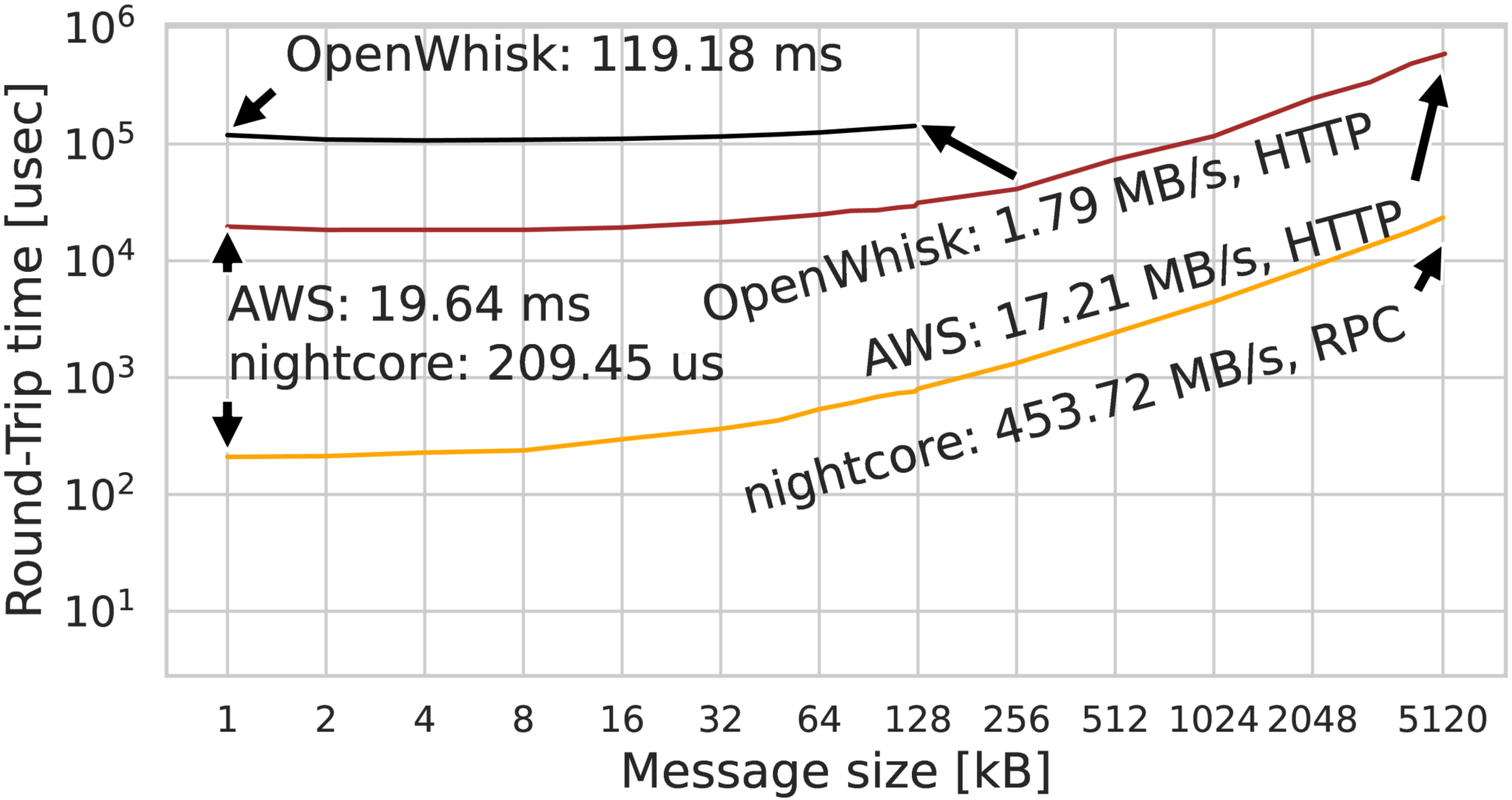
"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in FaaS?



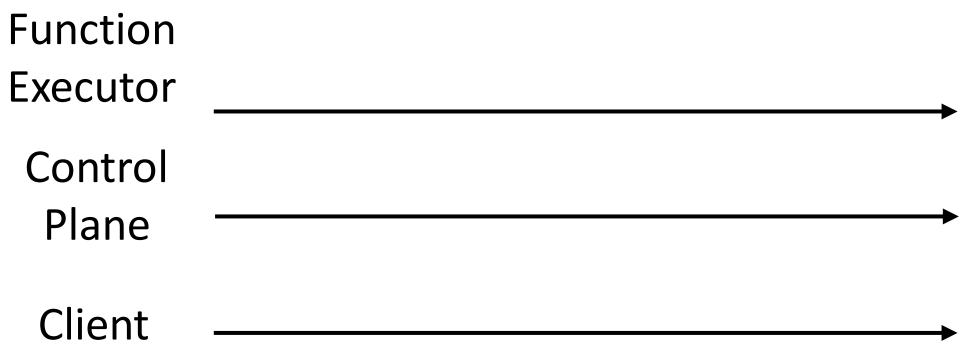
"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in FaaS?

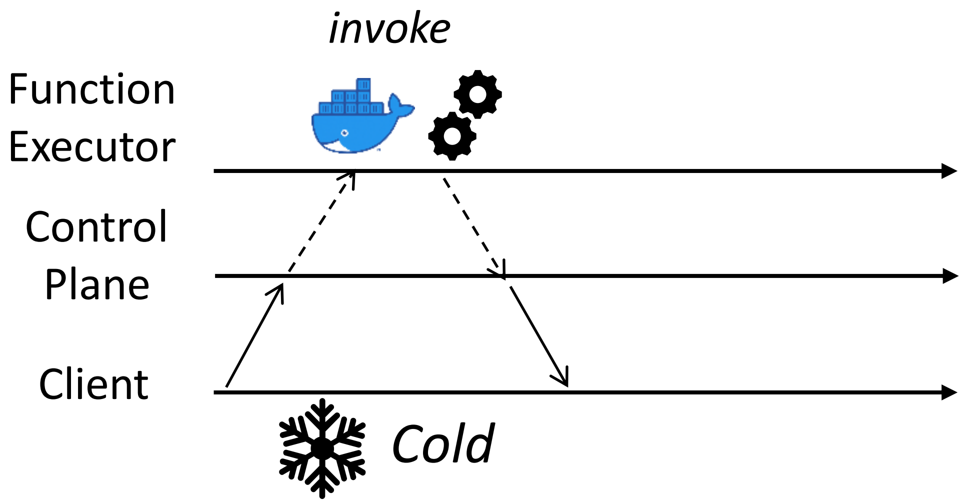


"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

Invocations in FaaS and rFaaS

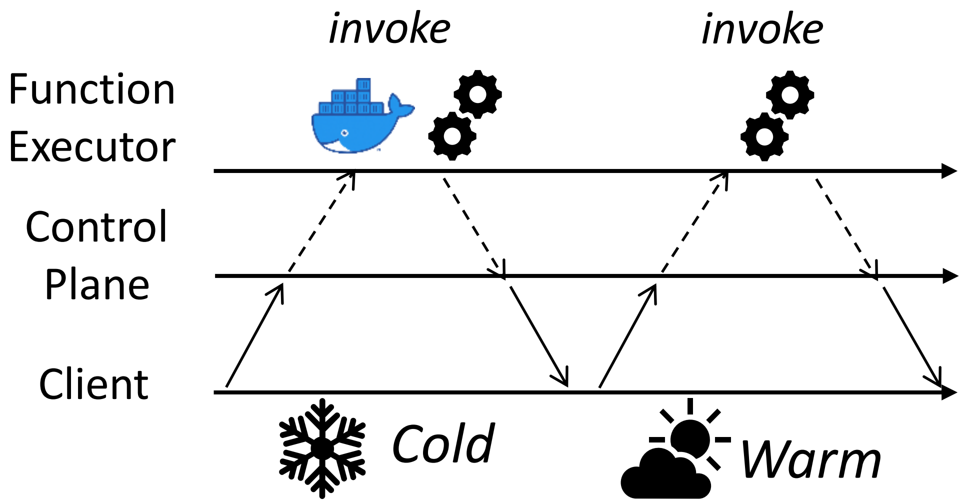


Invocations in FaaS and rFaaS



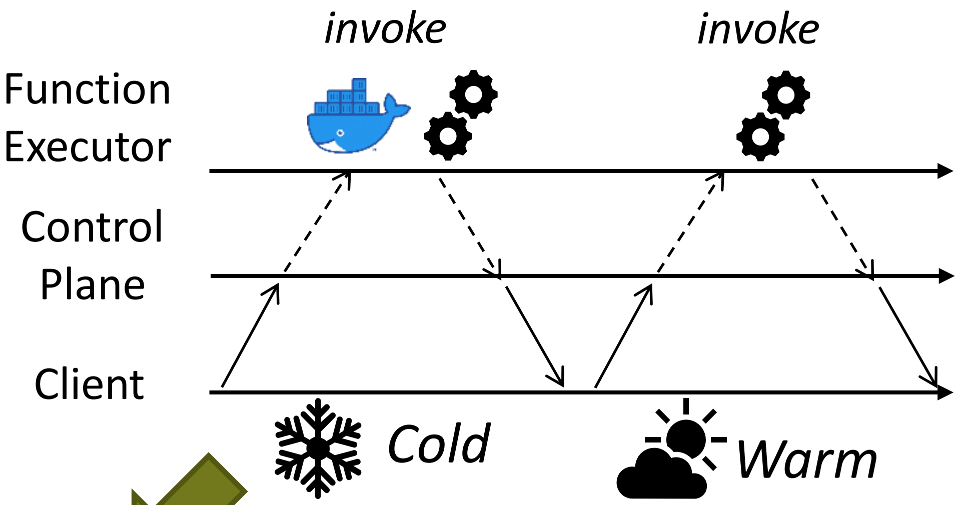
“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IPDPS’23

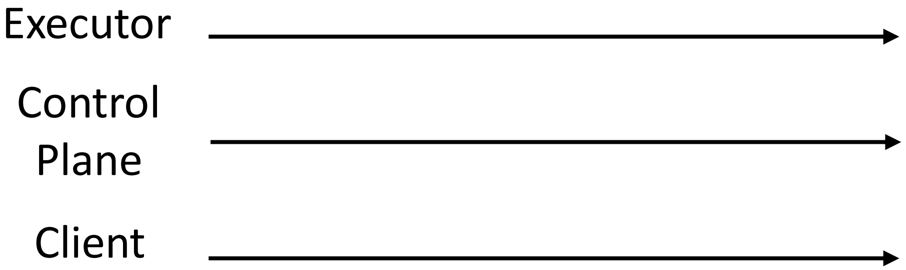
Invocations in FaaS and rFaaS



“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IPDPS’23

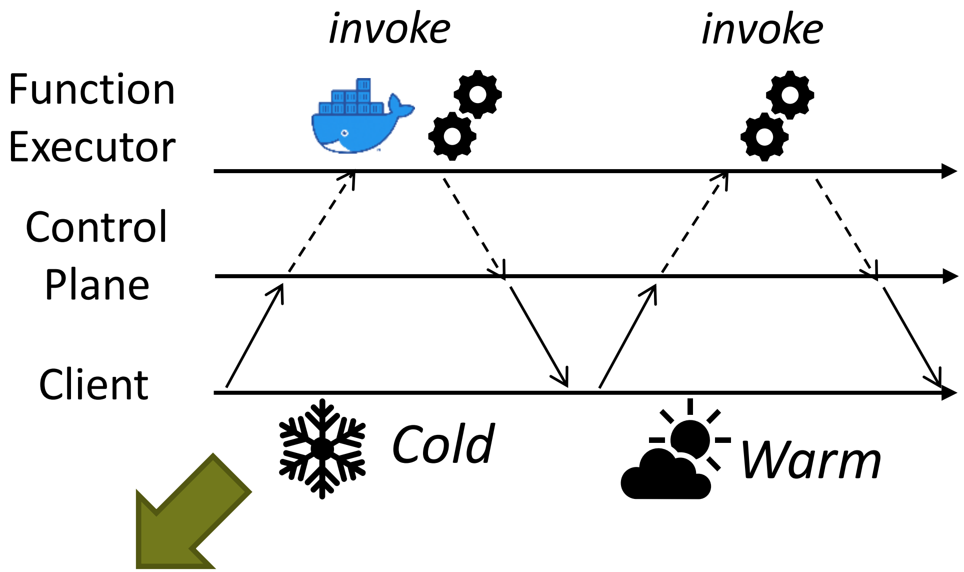
Invocations in FaaS and rFaaS

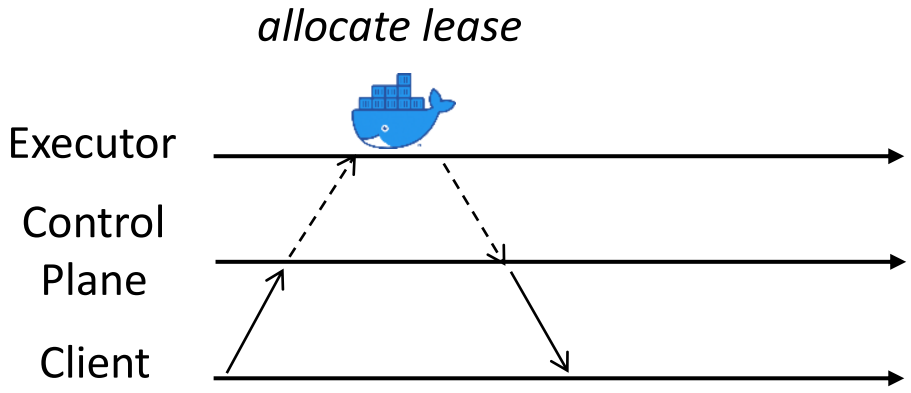




"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

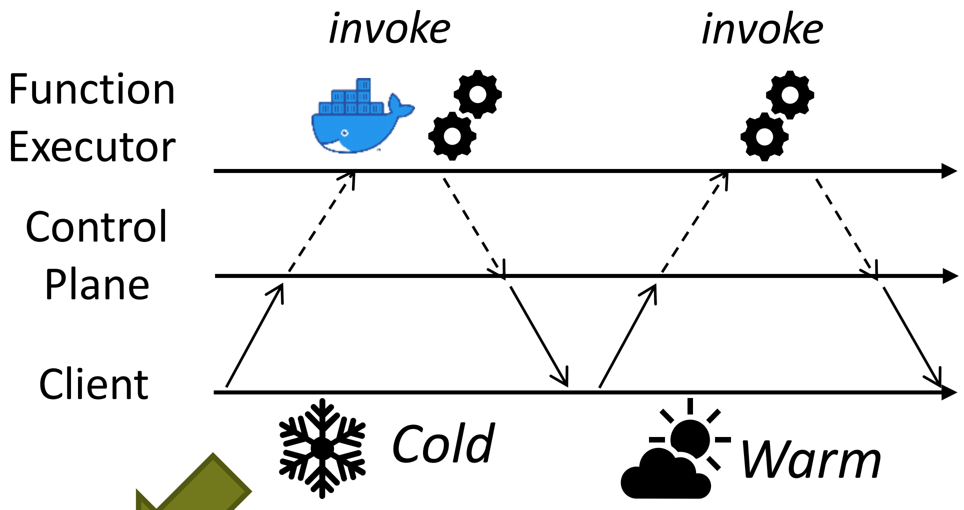
Invocations in FaaS and rFaaS

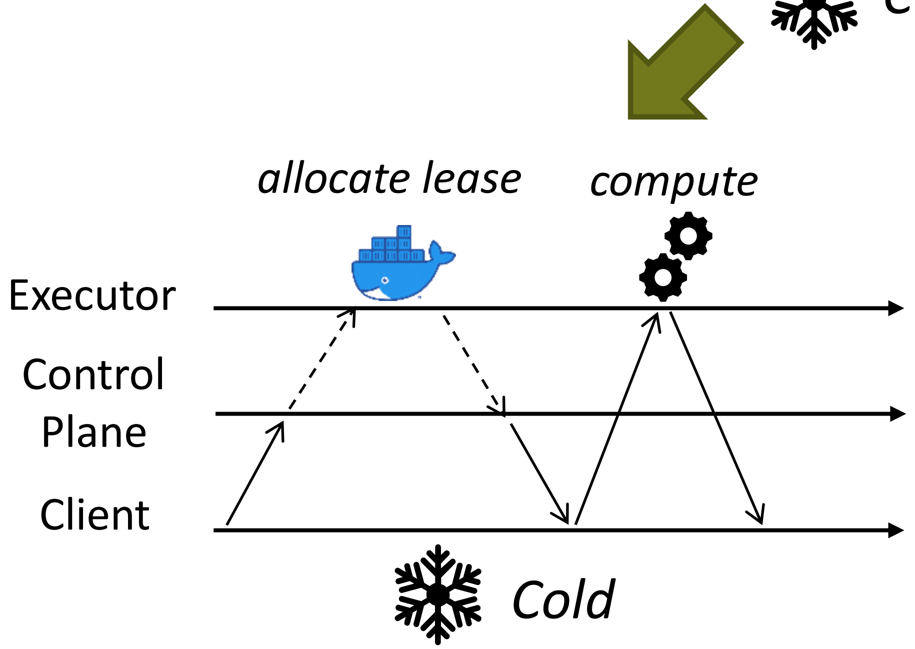




"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

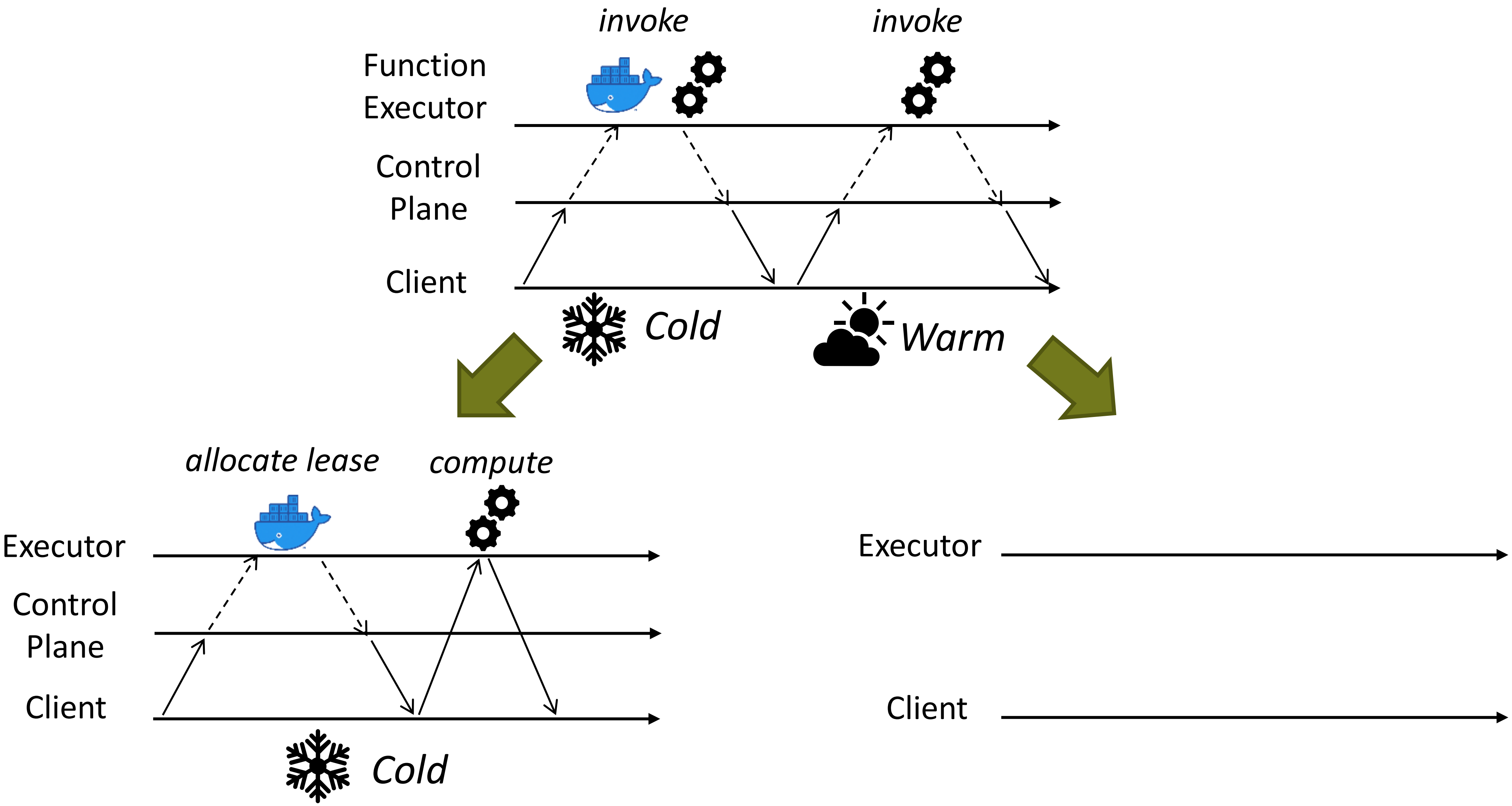
Invocations in FaaS and rFaaS





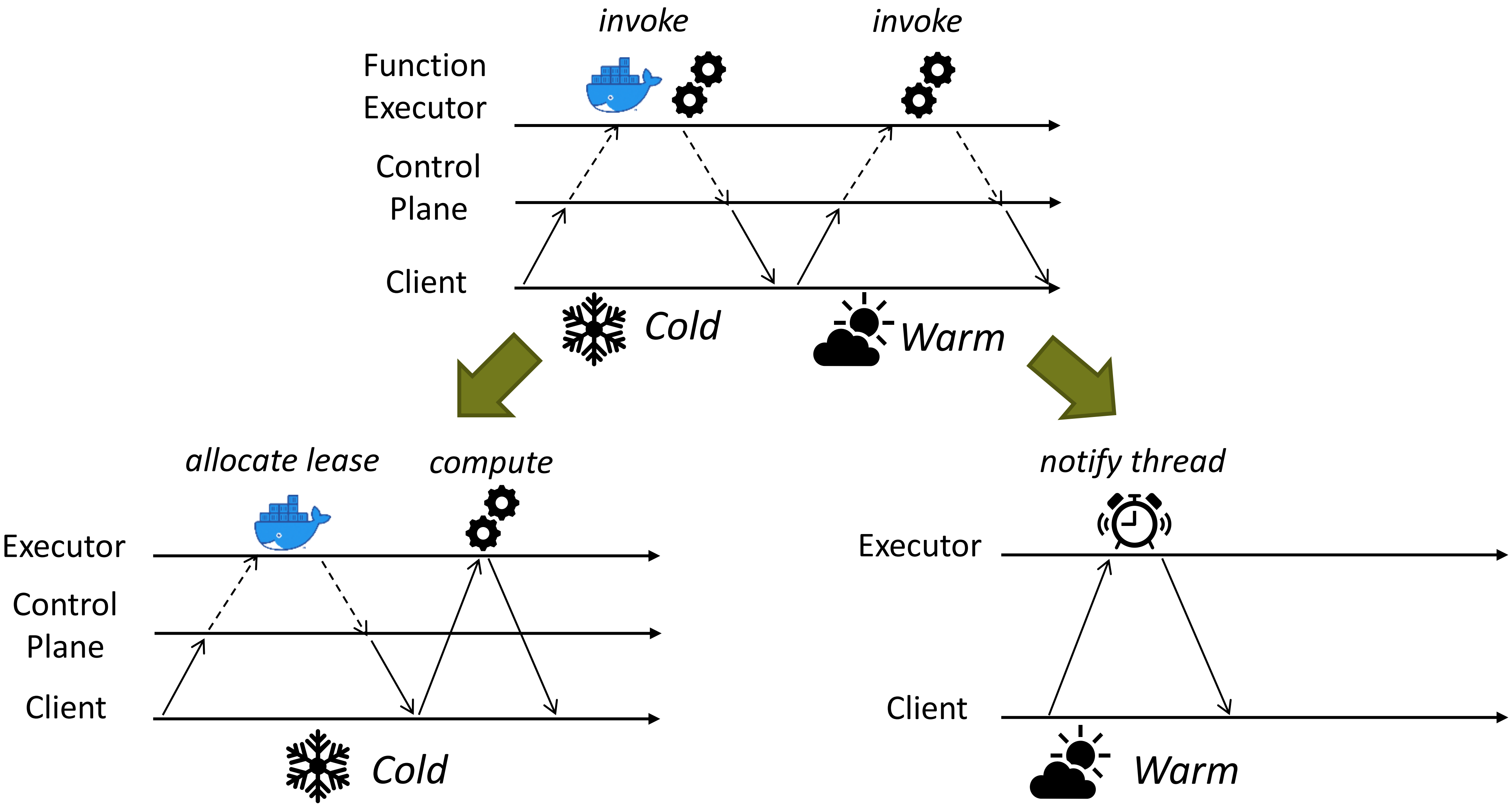
"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

Invocations in FaaS and rFaaS



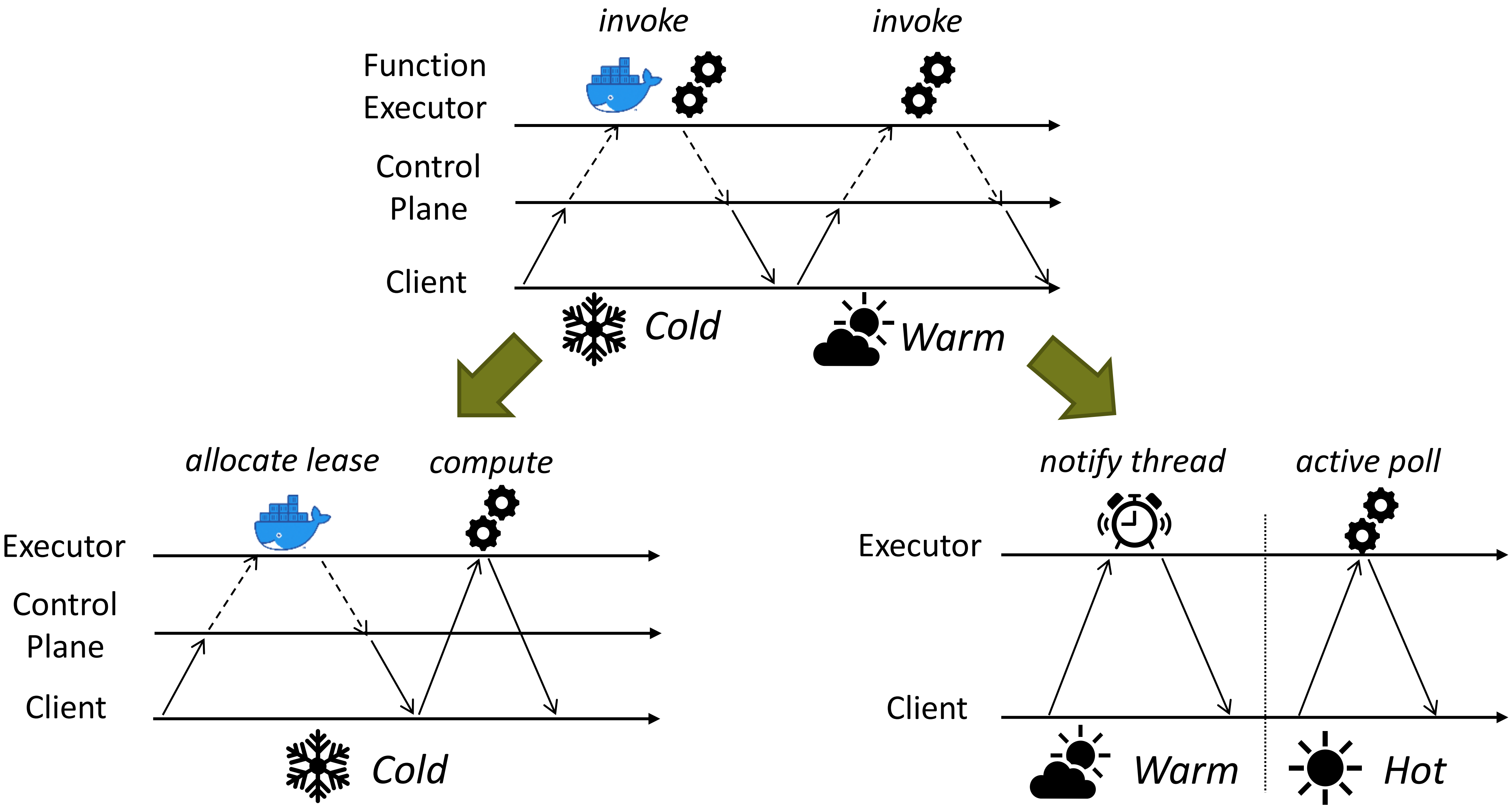
"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

Invocations in FaaS and rFaaS



“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IPDPS’23

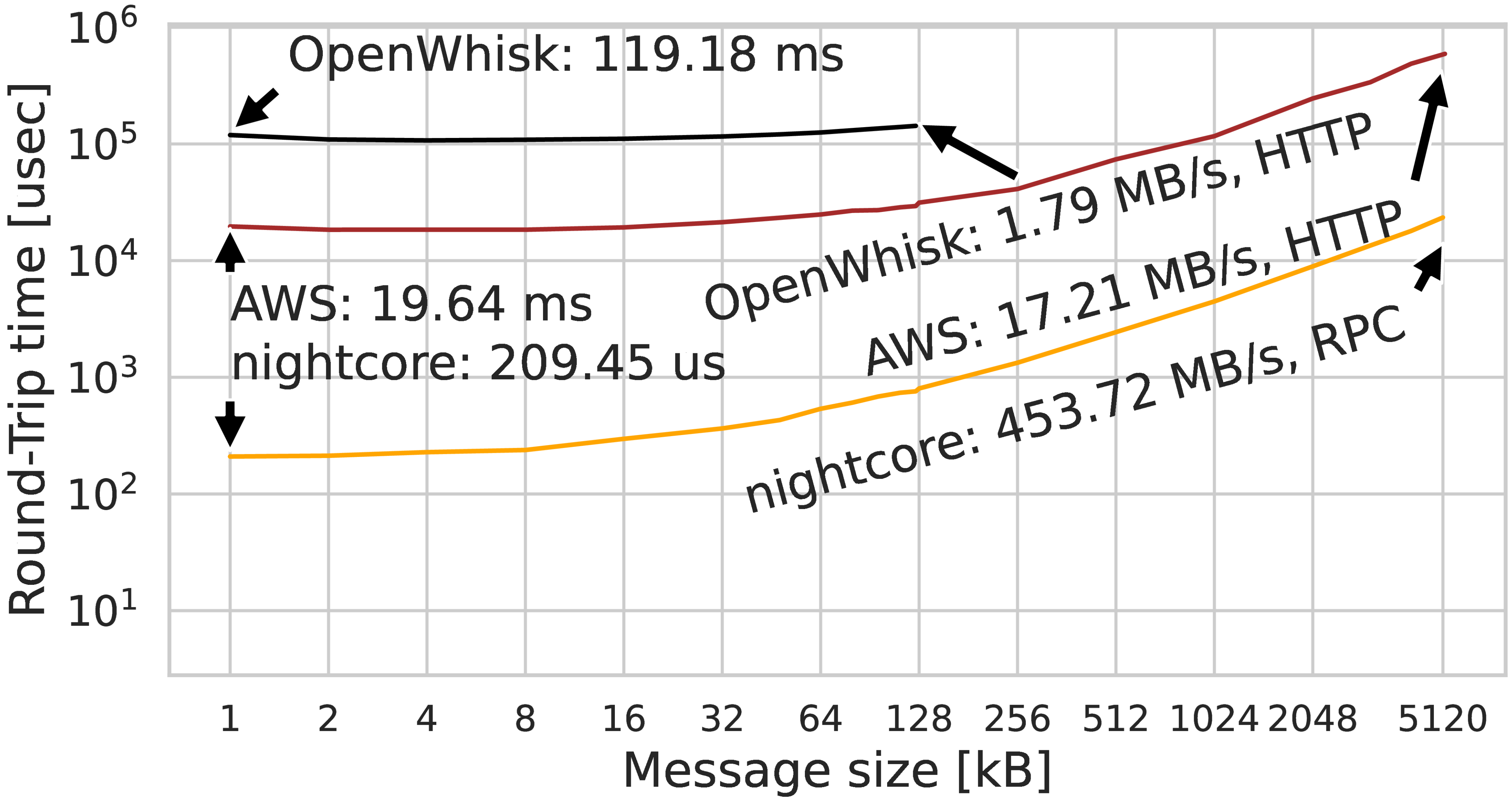
Invocations in FaaS and rFaaS



"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in rFaaS?

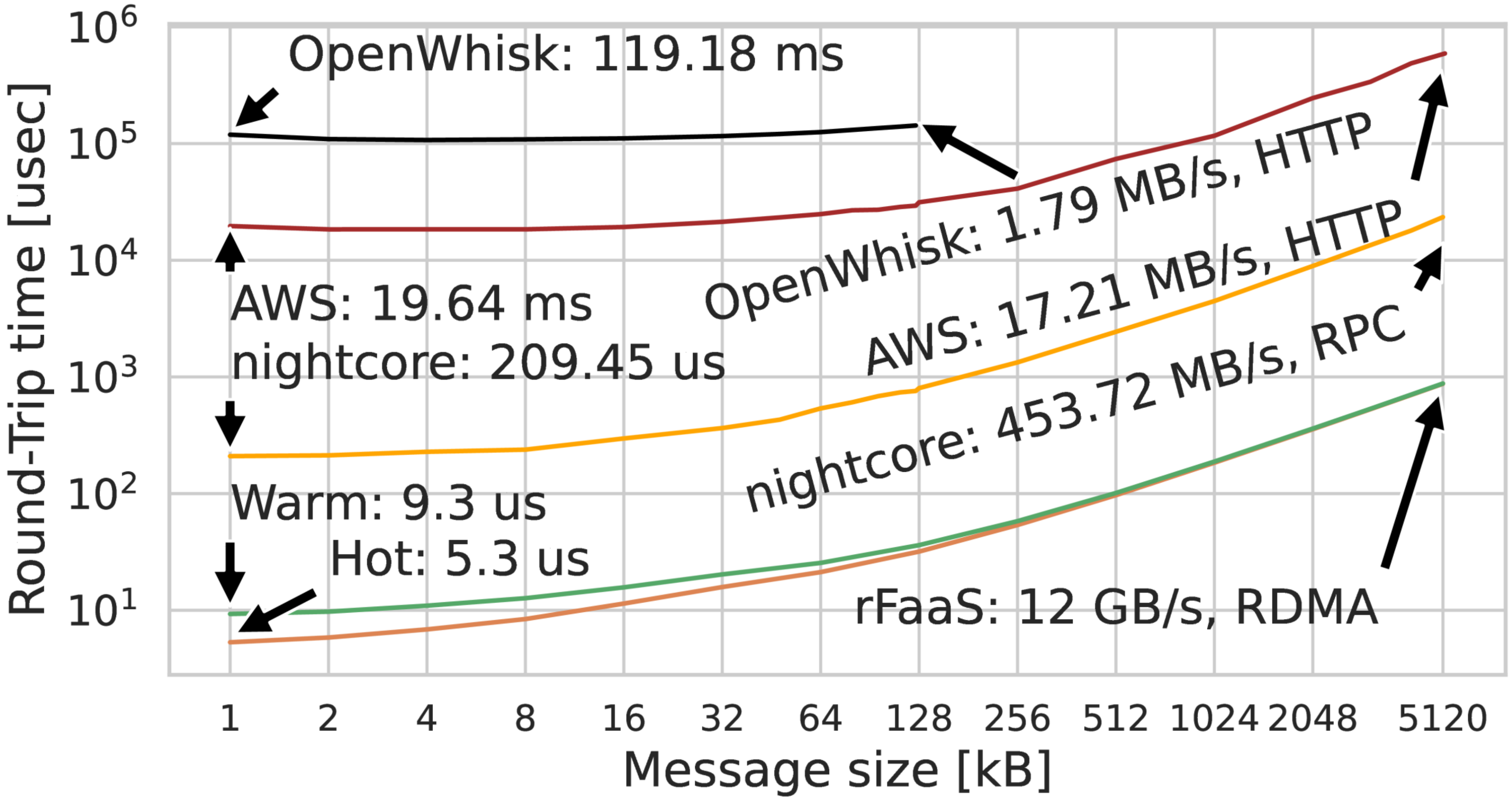
36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.



"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in rFaaS?

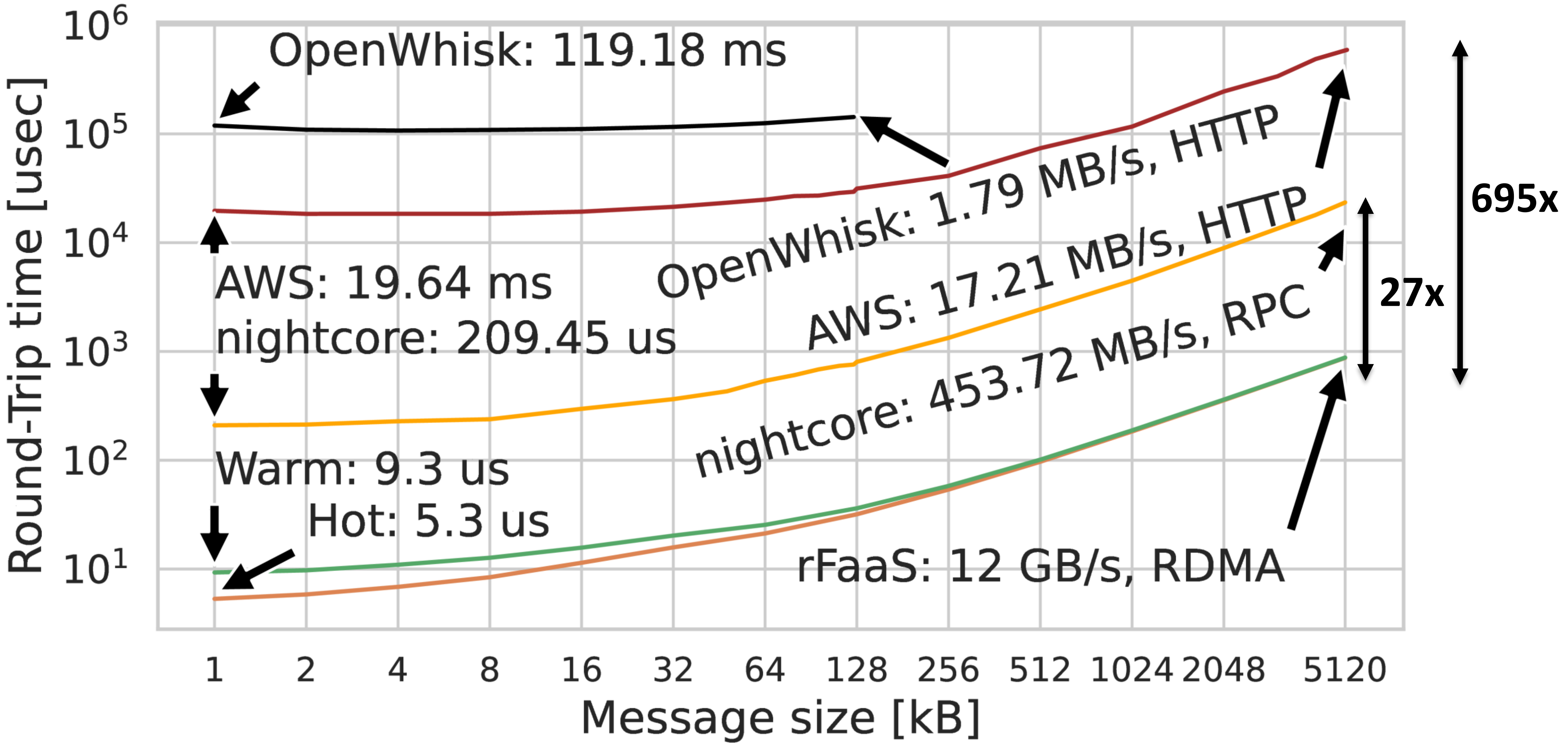
36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.



"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in rFaaS?

36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.



"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

FaaS in High-Performance Applications

Serverless is slow

Communication is slow and restricted

Answer: rFaaS

Serverless is hard to program.

FaaS in High-Performance Applications

Serverless is slow

Answer:
rFaaS

Serverless is hard to program.

Communication is slow and restricted

Answer:
FMI

Communication in serverless

"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

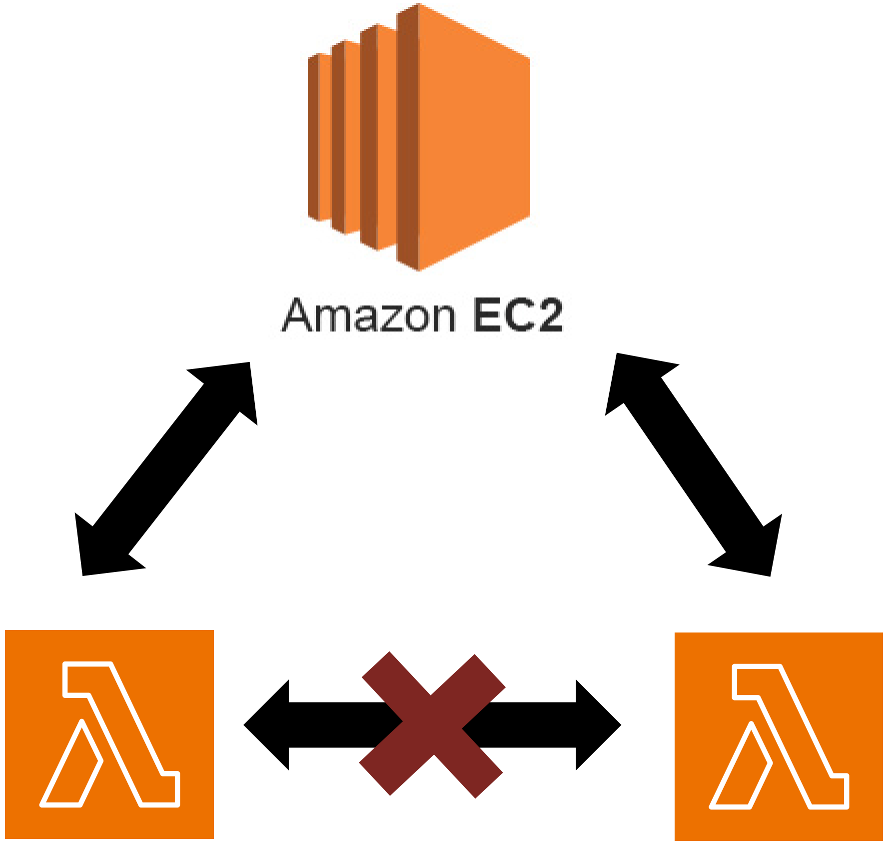
Communication in serverless

“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless

“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless

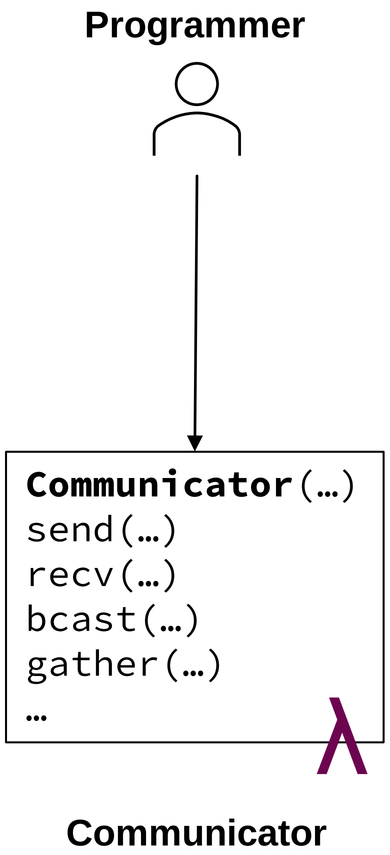


"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FMI: MPI for serverless

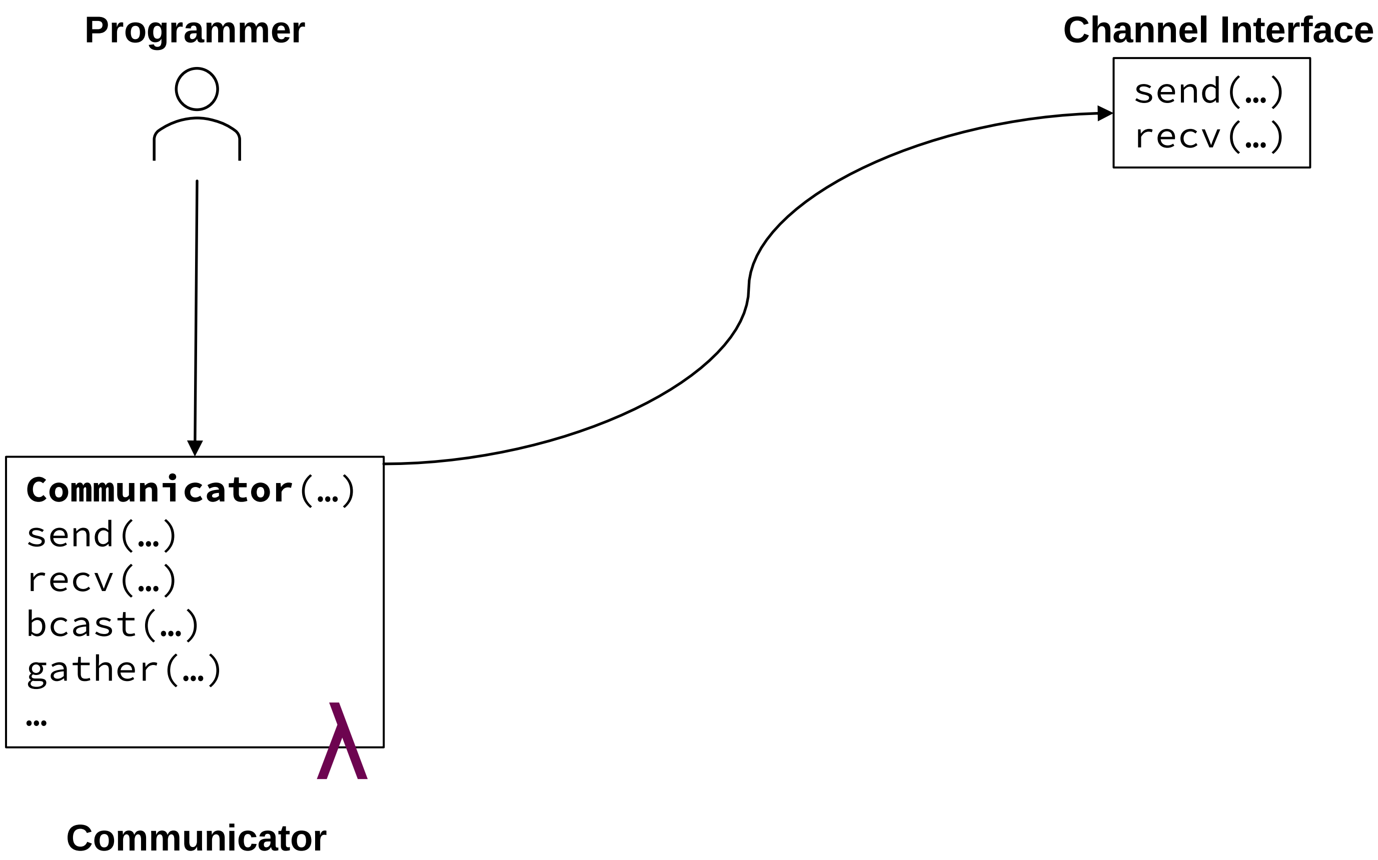
"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FMI: MPI for serverless



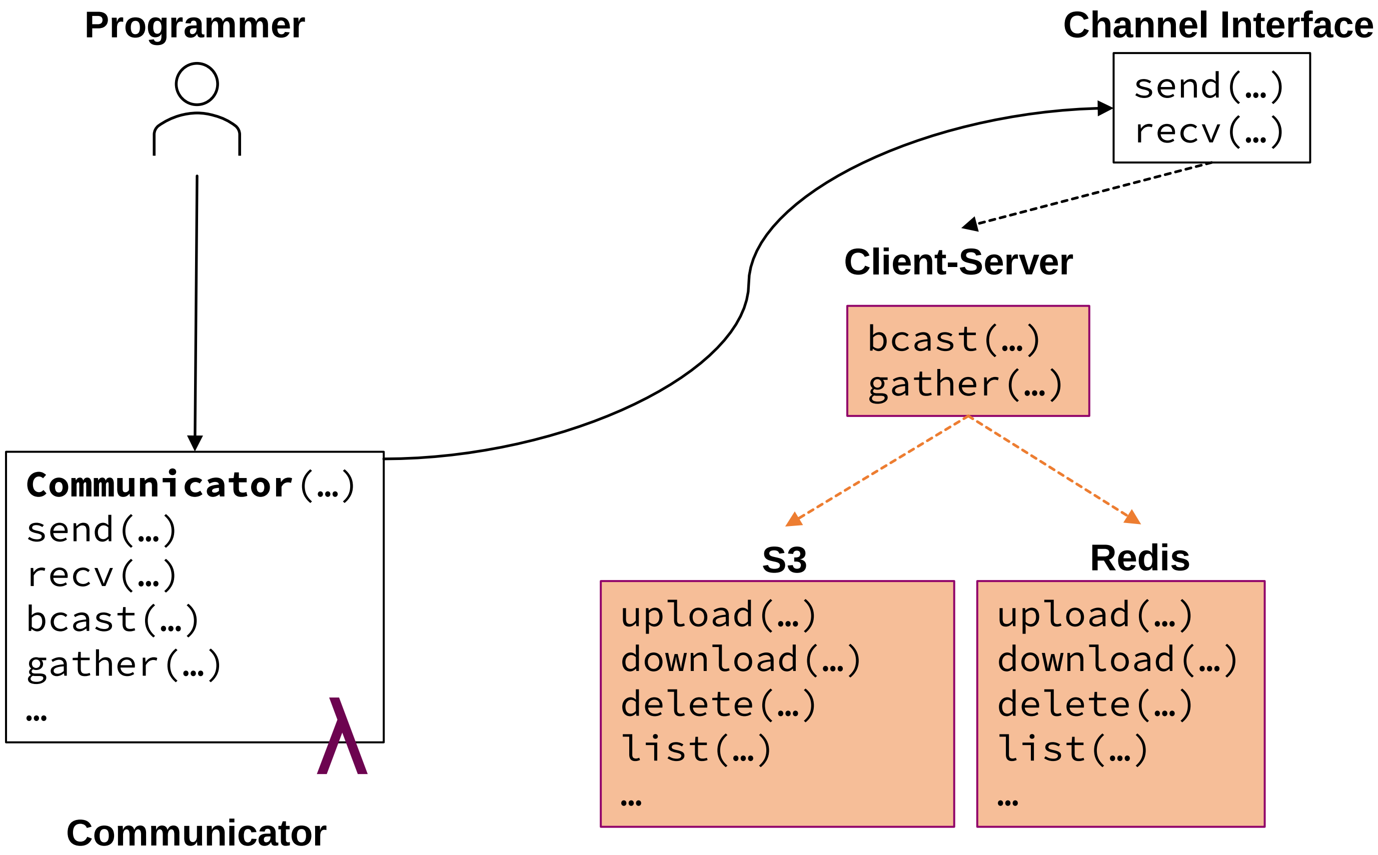
"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FMI: MPI for serverless



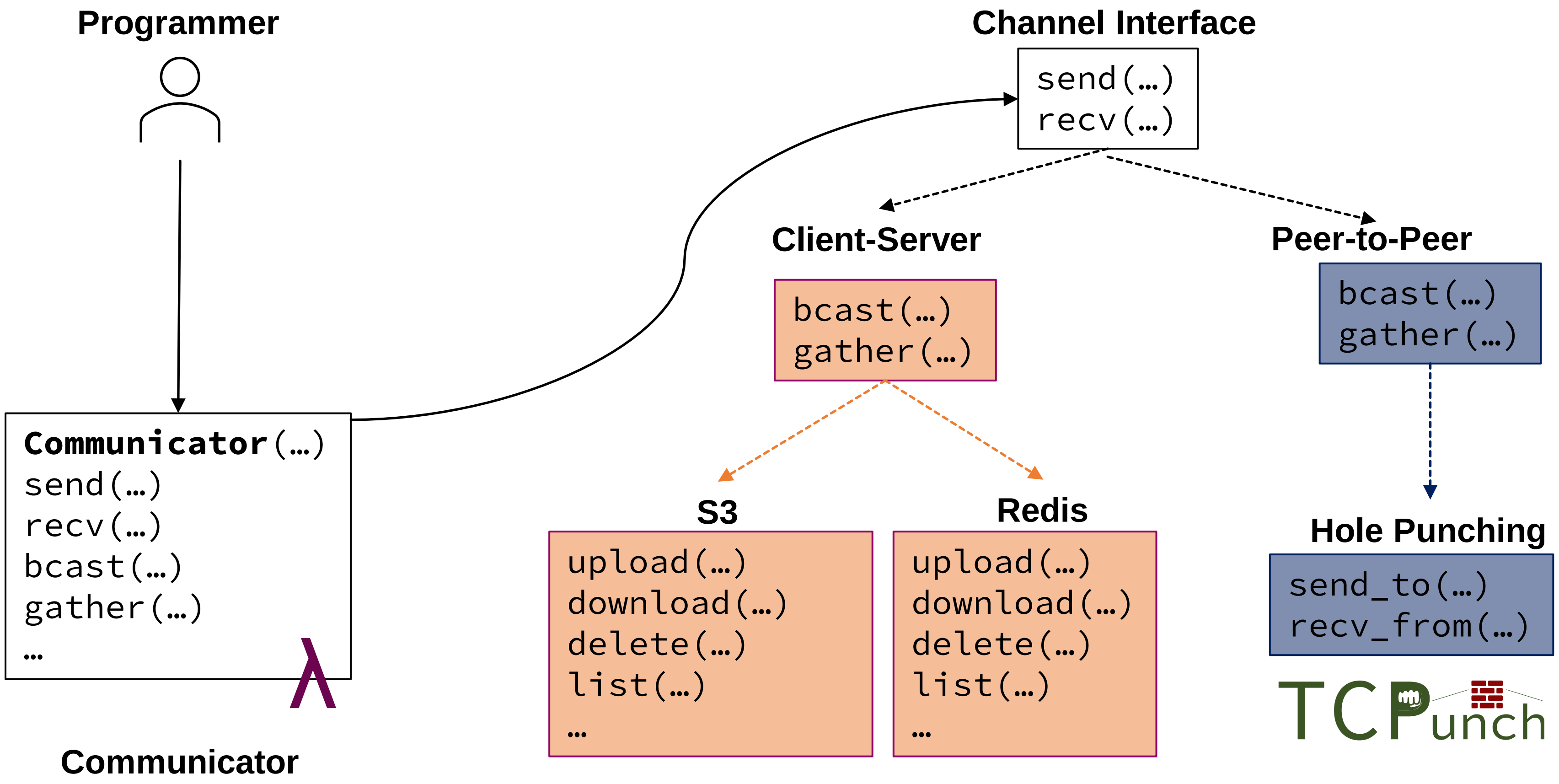
"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FMI: MPI for serverless



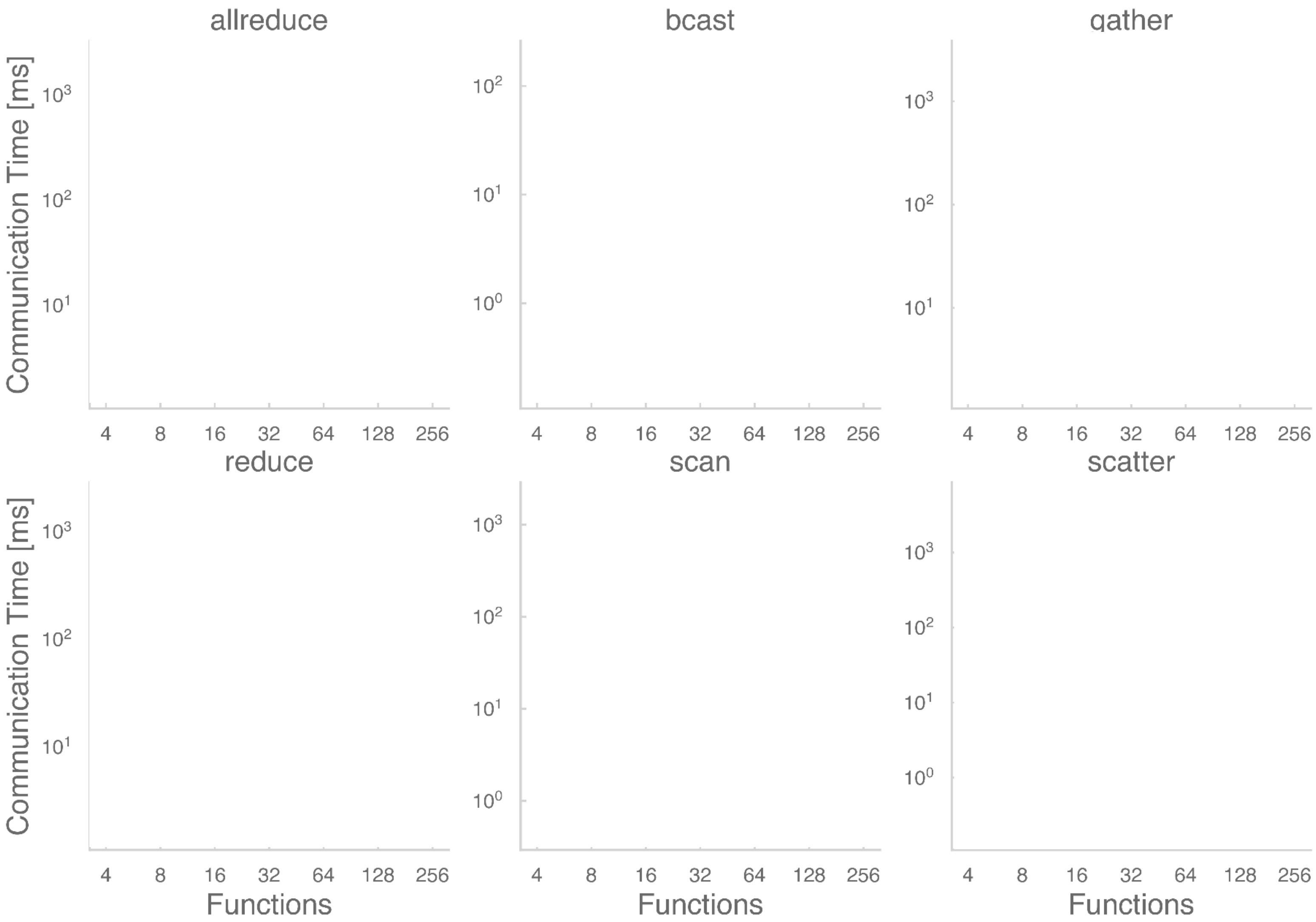
"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FMI: MPI for serverless



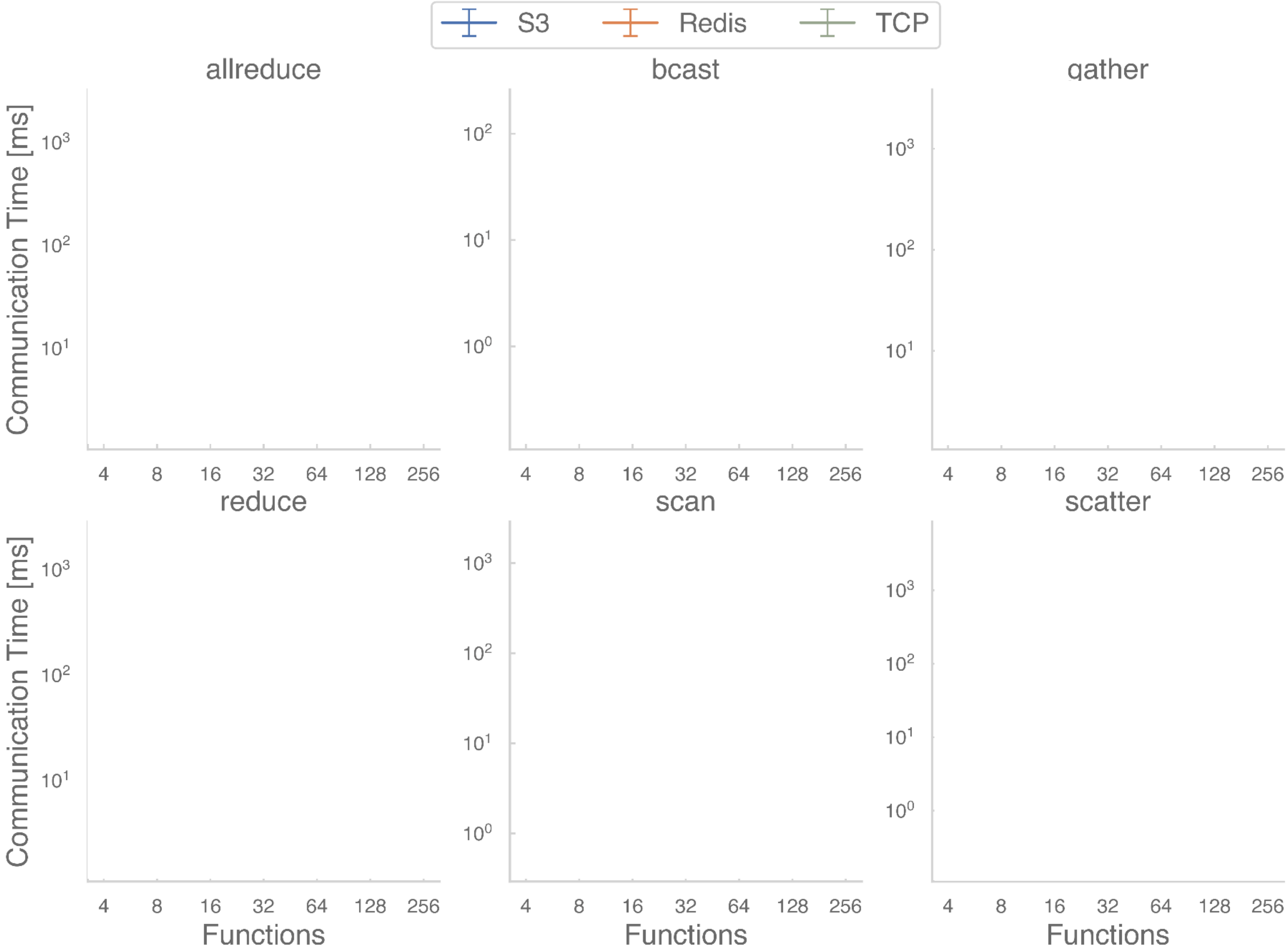
"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FMI on AWS Lambda



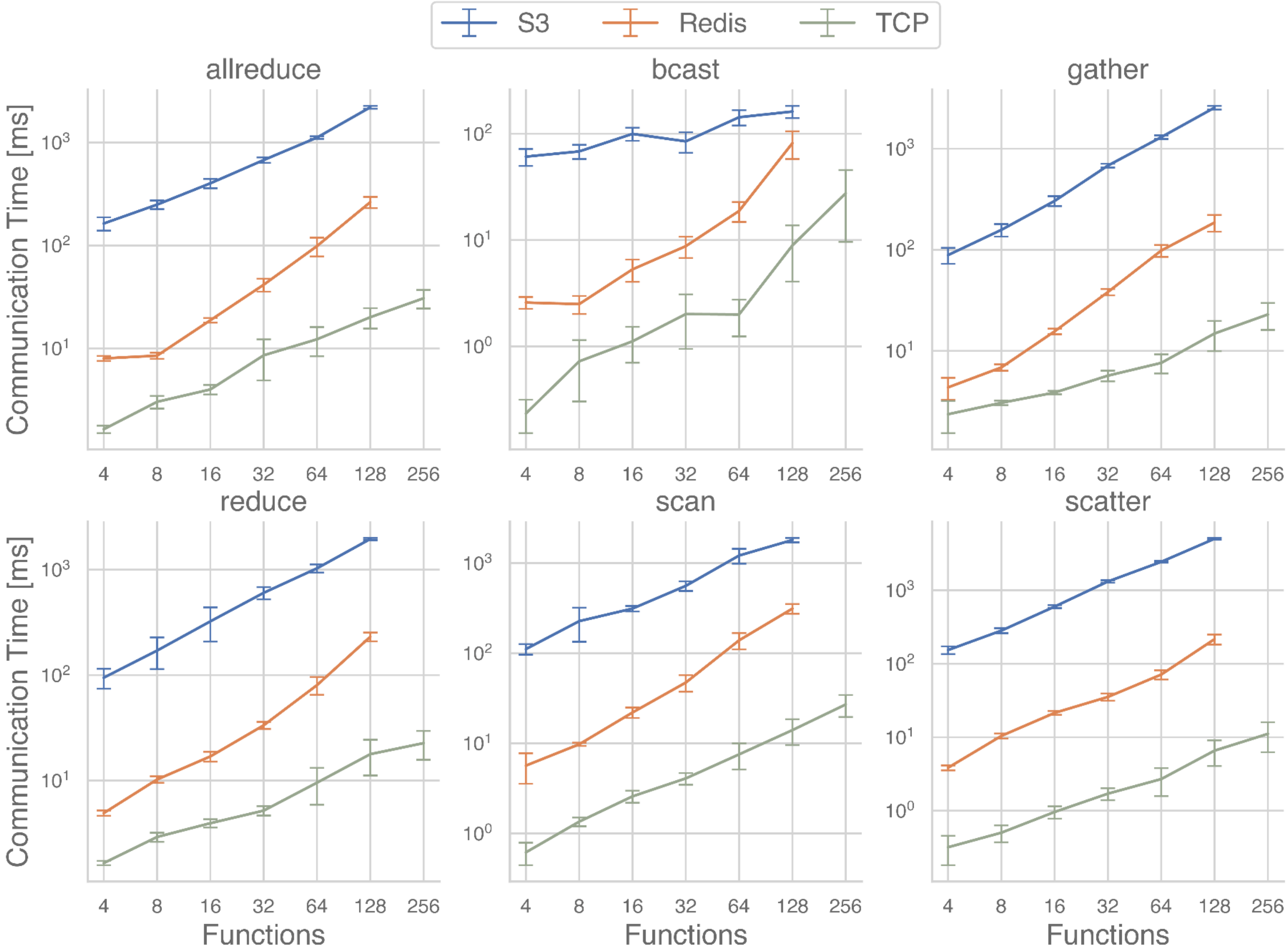
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

FMI on AWS Lambda



"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FMI on AWS Lambda



"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

FaaS in High-Performance Applications

Serverless is slow

Communication is slow and restricted

Answer: rFaaS

Answer: FMI

Serverless is hard to program.

FaaS in High-Performance Applications

Serverless is slow

Answer:
rFaaS

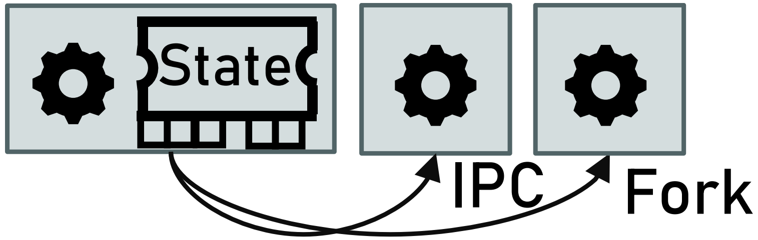
Communication is slow
and restricted

Answer:
FMI

Serverless is hard to
program.

Answer: Serverless
Processes

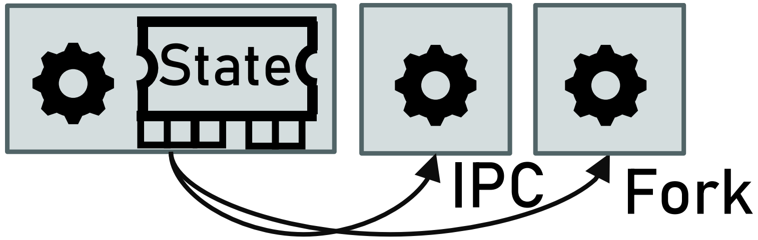
Serverless Process



OS Process

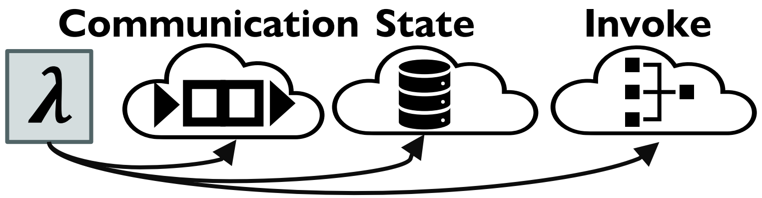
Nano- and micro-second latency of OS primitives.

Serverless Process



OS Process

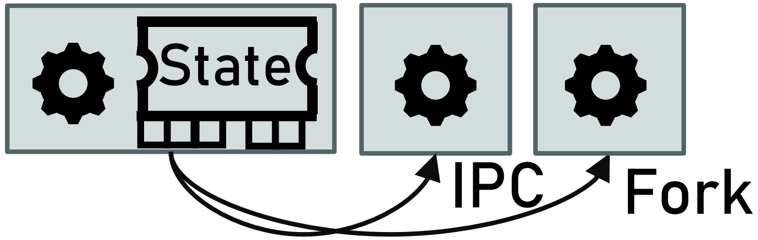
Nano- and micro-second latency of OS primitives.



Serverless Function

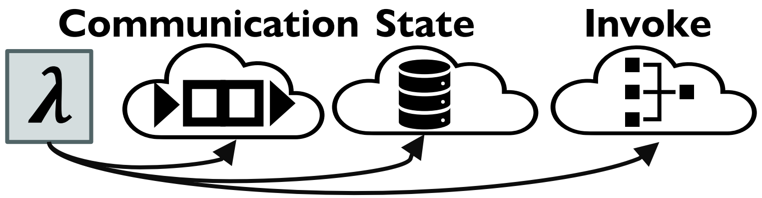
Millisecond latency of cloud proxies.

Serverless Process



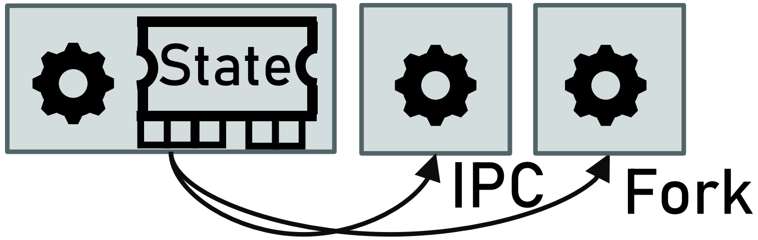
OS Process
Nano- and micro-second latency of OS primitives.

+



Serverless Function
Millisecond latency of cloud proxies.

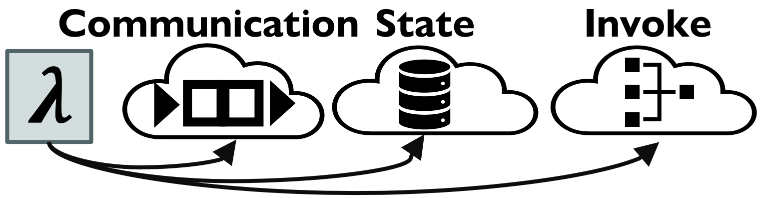
Serverless Process



OS Process

Nano- and micro-second latency of OS primitives.

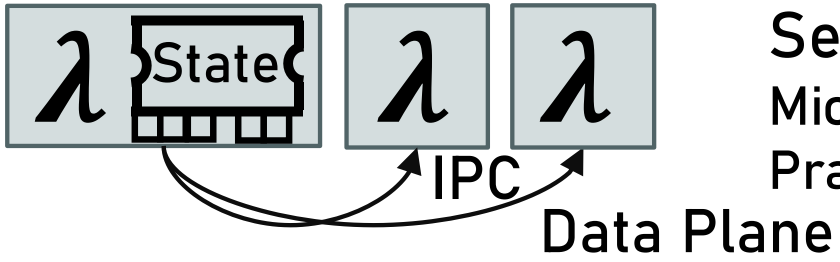
+



Serverless Function

Millisecond latency of cloud proxies.

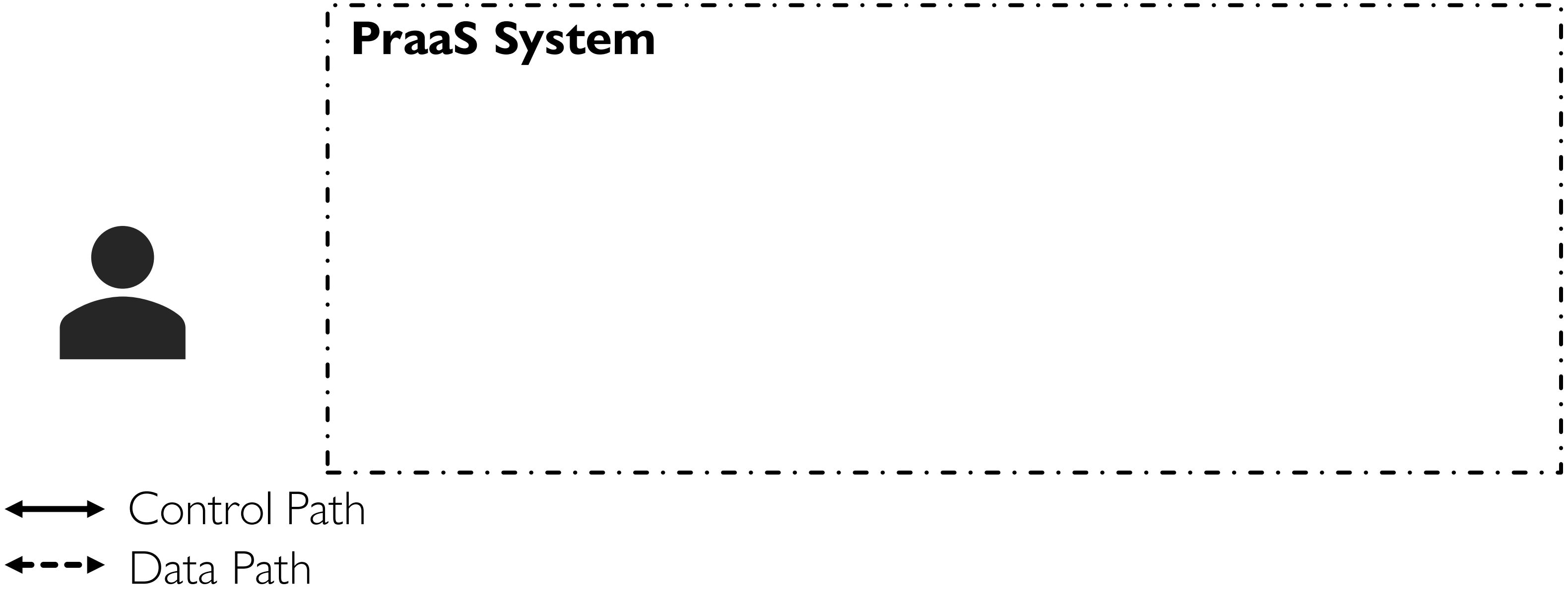
=



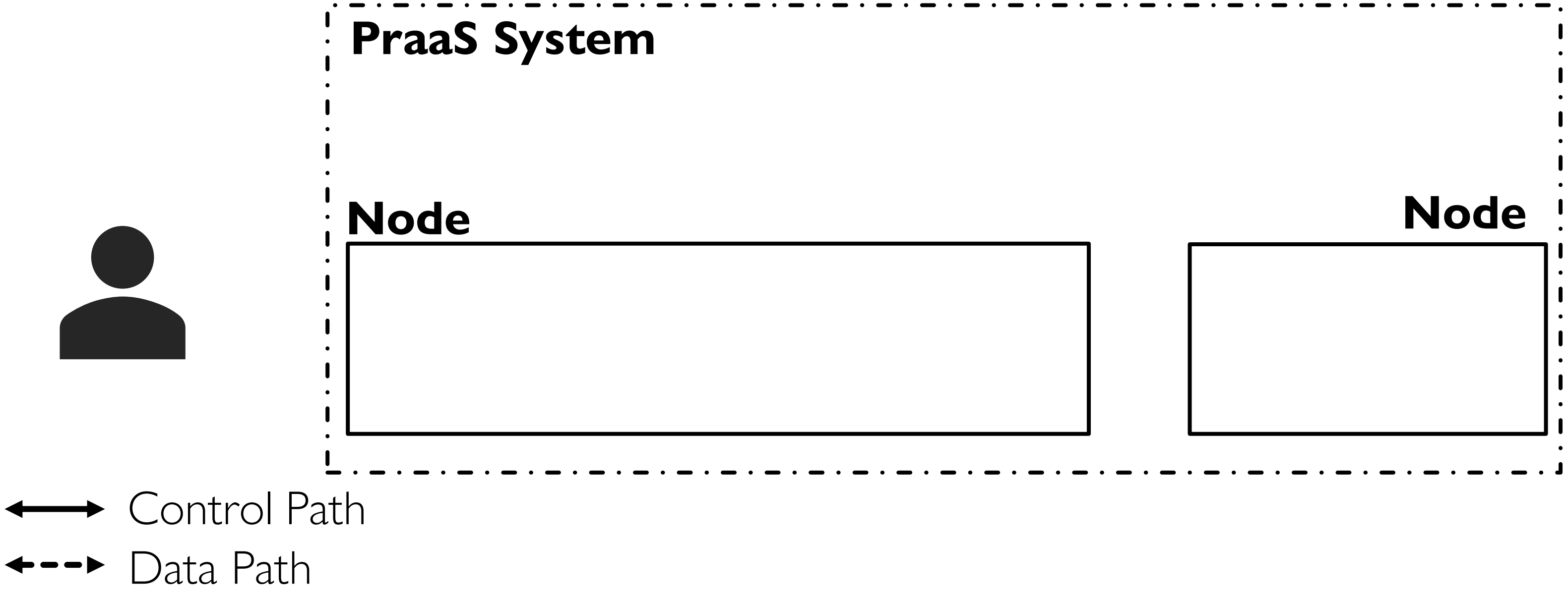
Serverless Process

Microsecond latency of PraaS backend.

PraaS: Process-as-Service

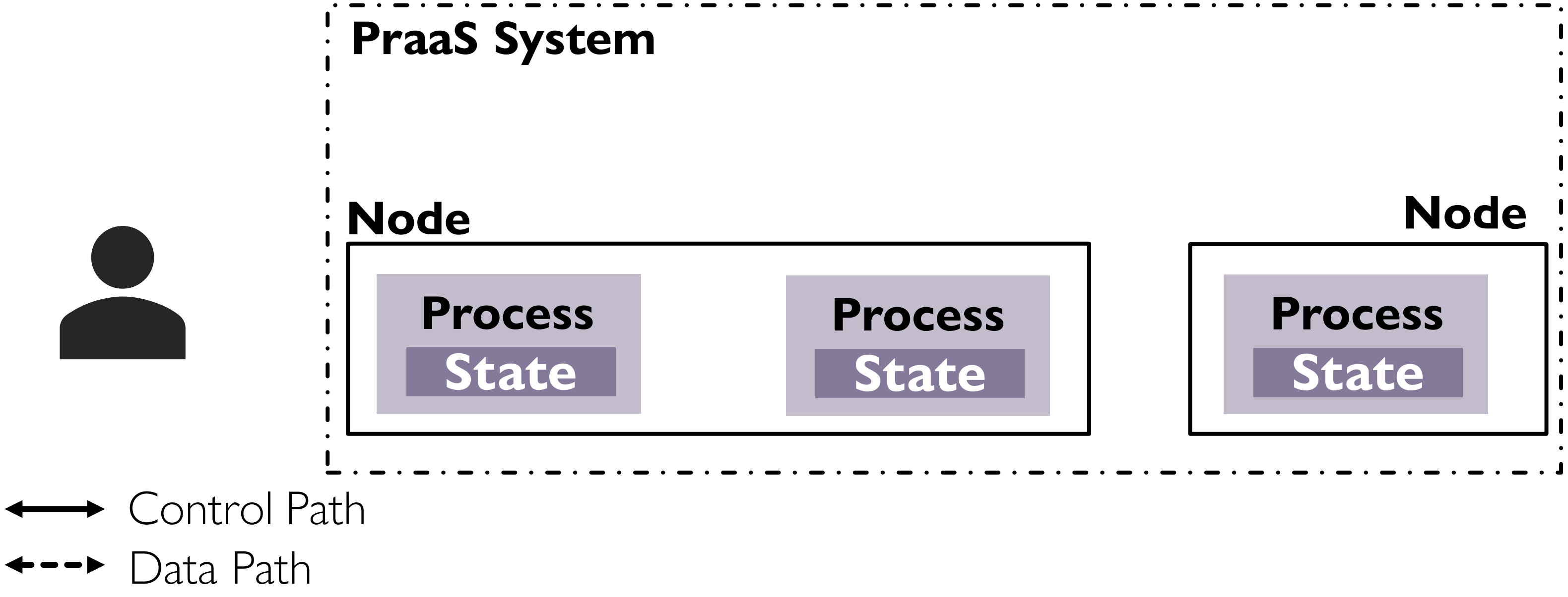


PraaS: Process-as-Service



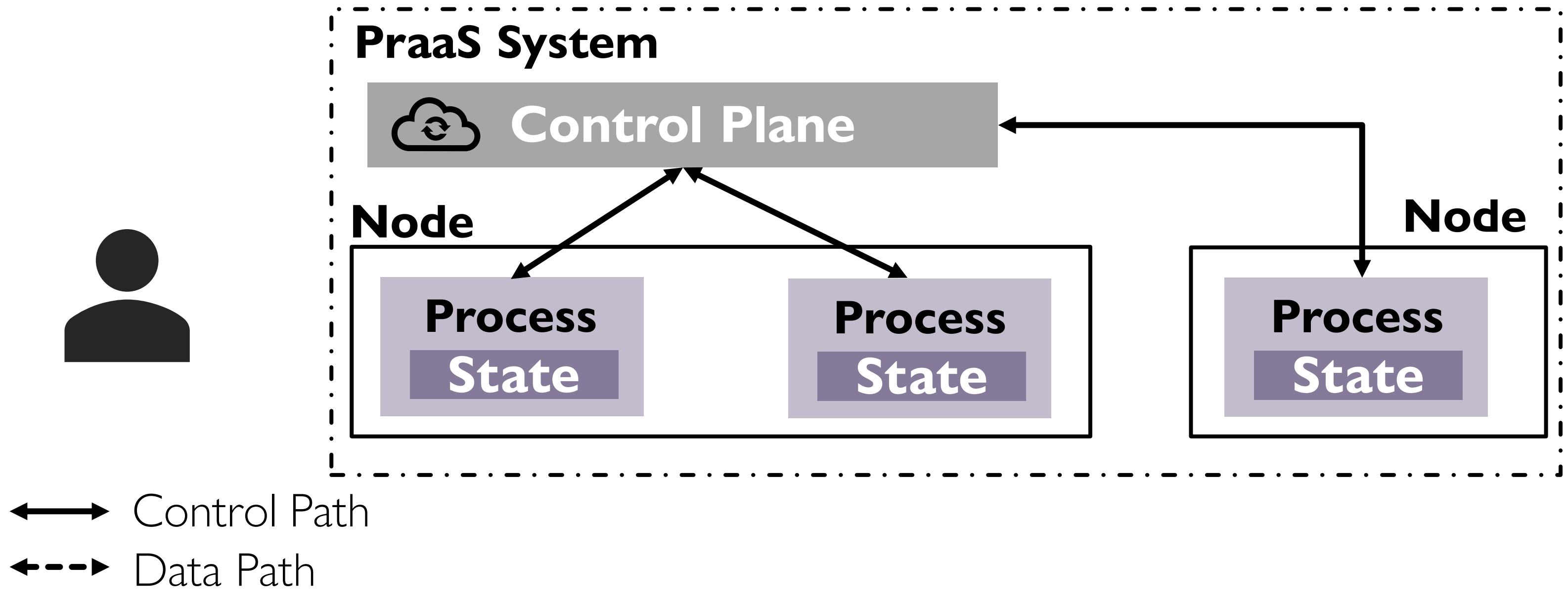
"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

PraaS: Process-as-Service



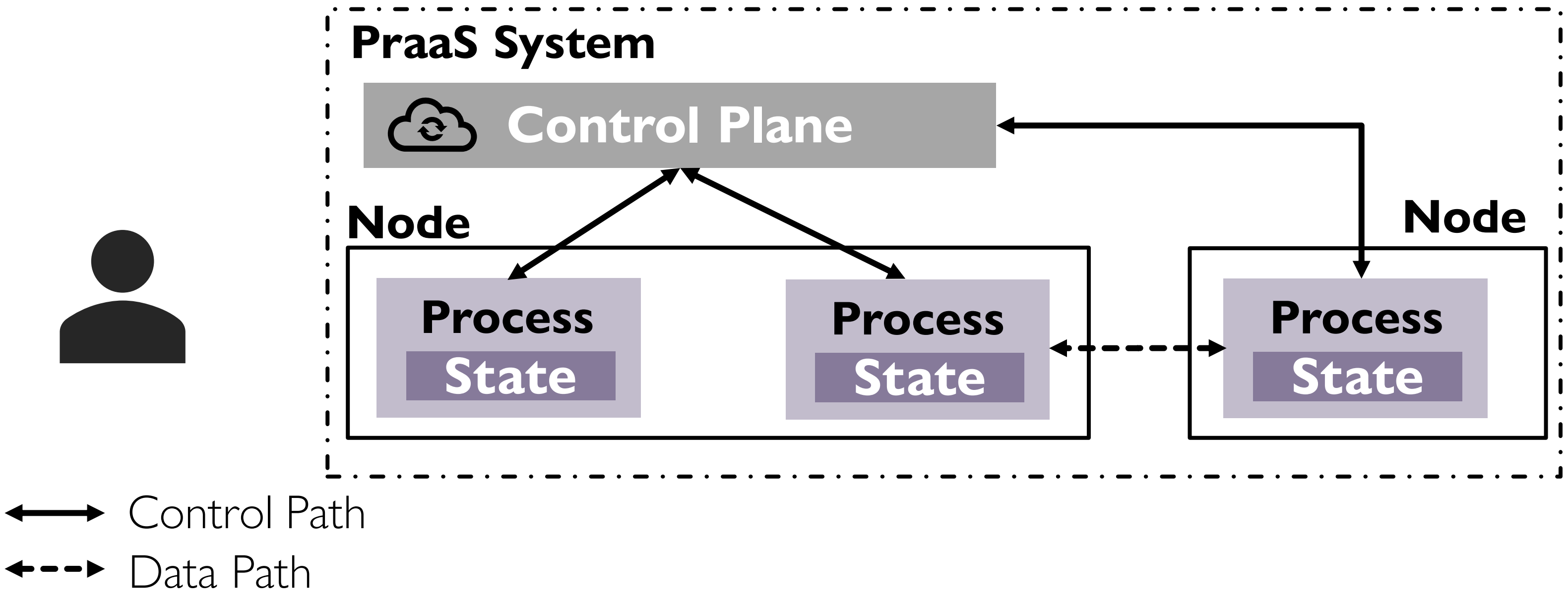
"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

PraaS: Process-as-Service



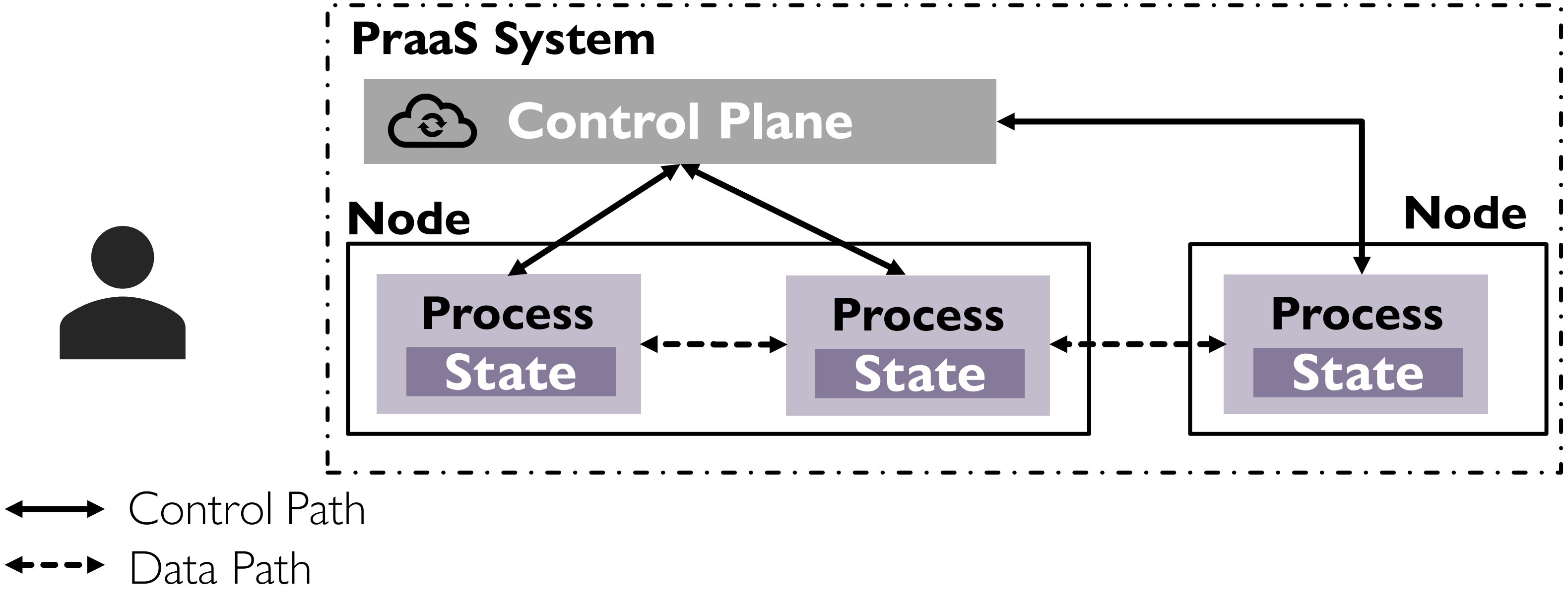
“Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes”, paper preprint.

PraaS: Process-as-Service



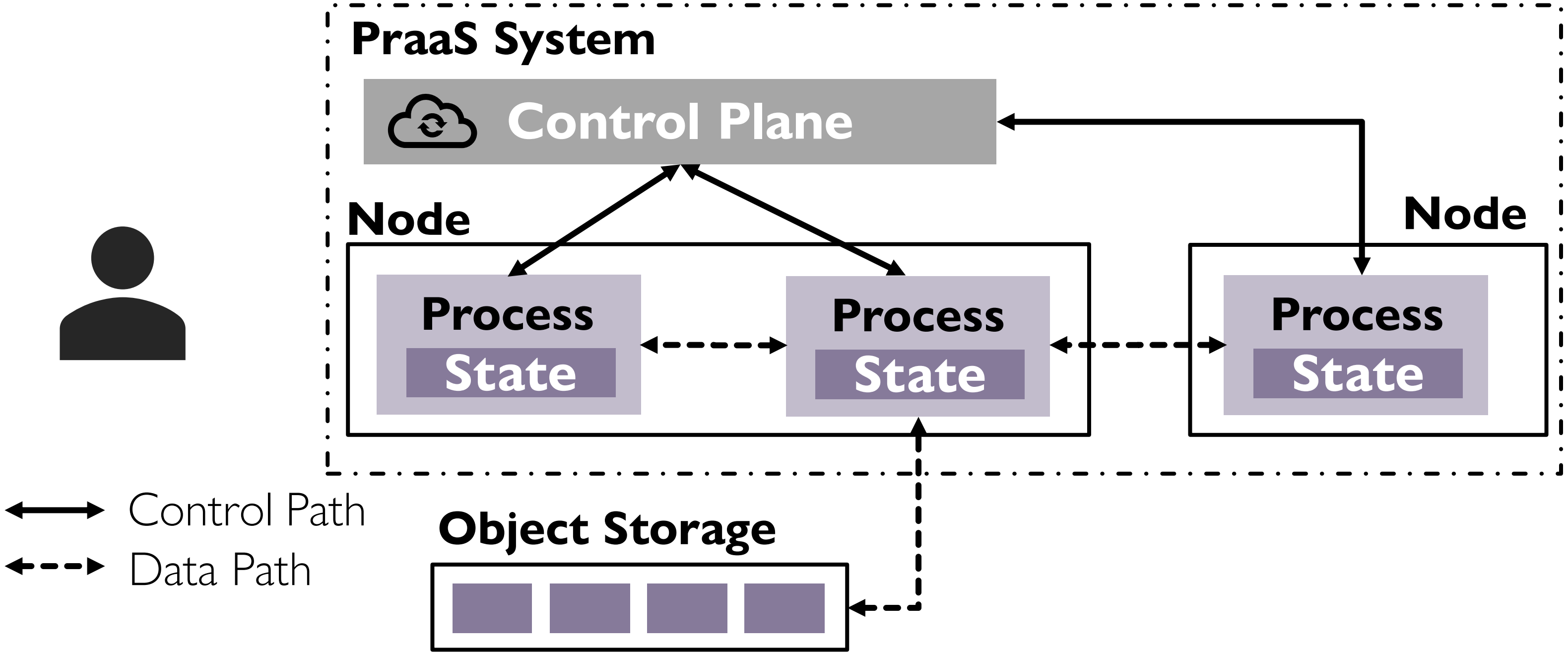
"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

PraaS: Process-as-Service



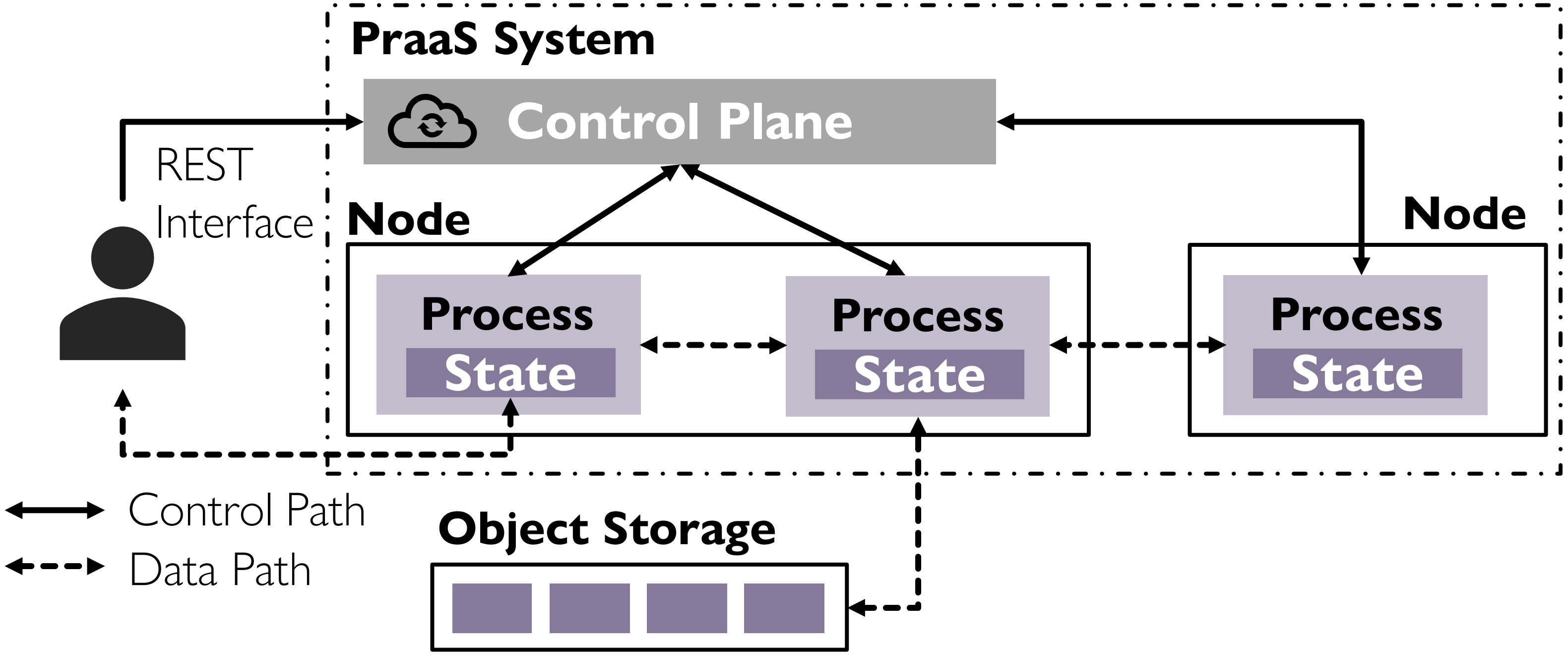
“Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes”, paper preprint.

PraaS: Process-as-Service



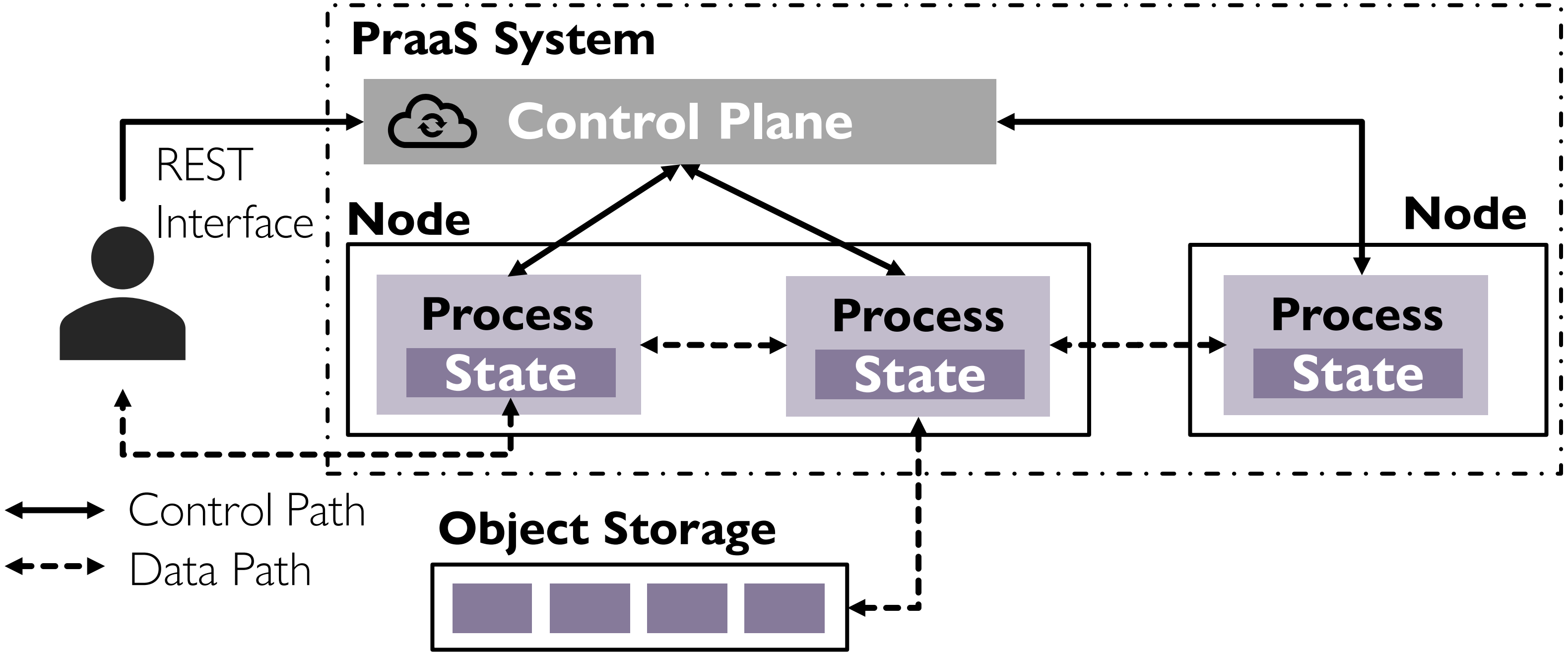
"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

PraaS: Process-as-Service



“Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes”, paper preprint.

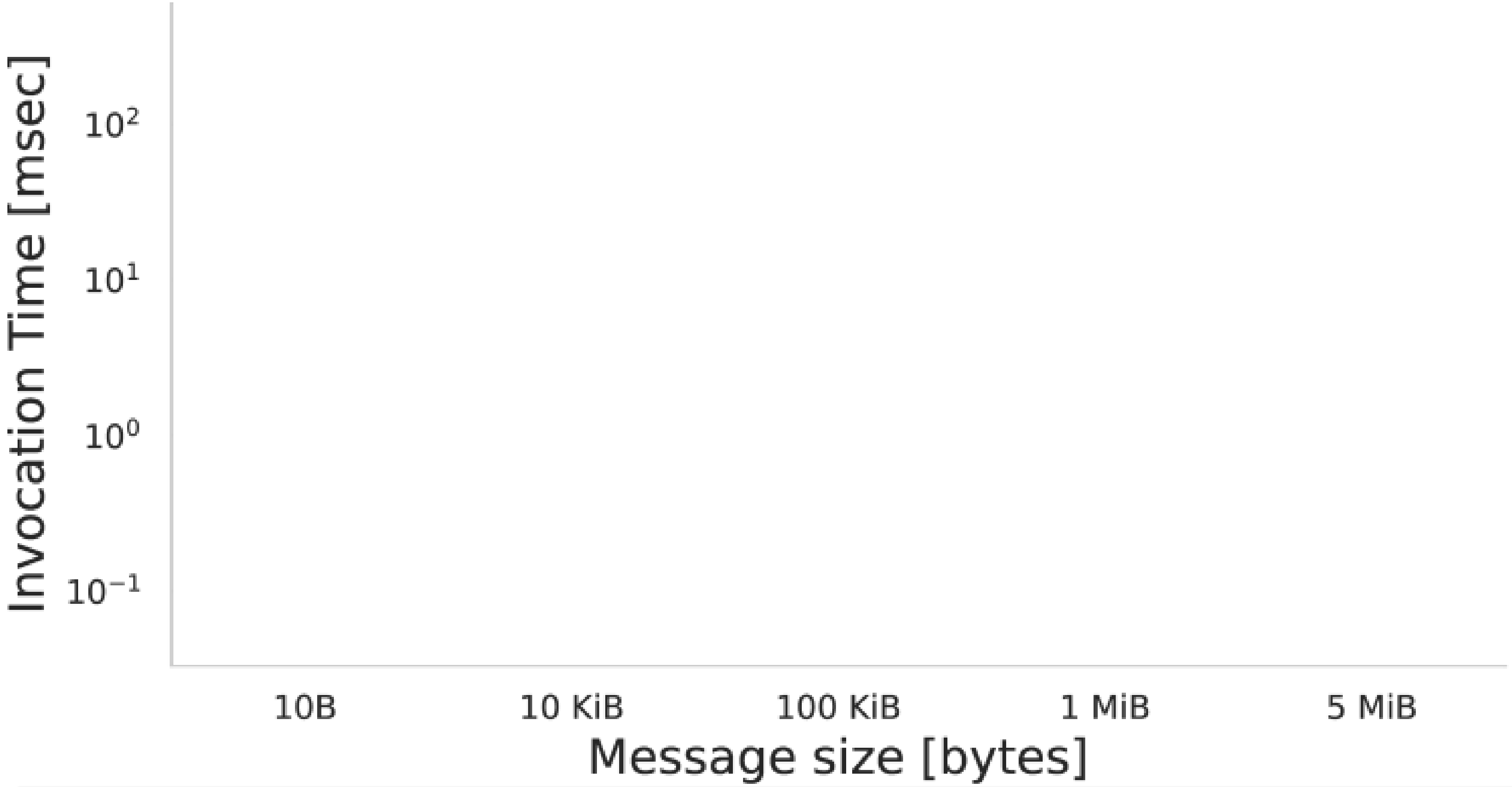
PraaS: Process-as-Service



Works on AWS Fargate, Knative, Kubernetes.

"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

Serverless Process on Fargate vs AWS Lambda

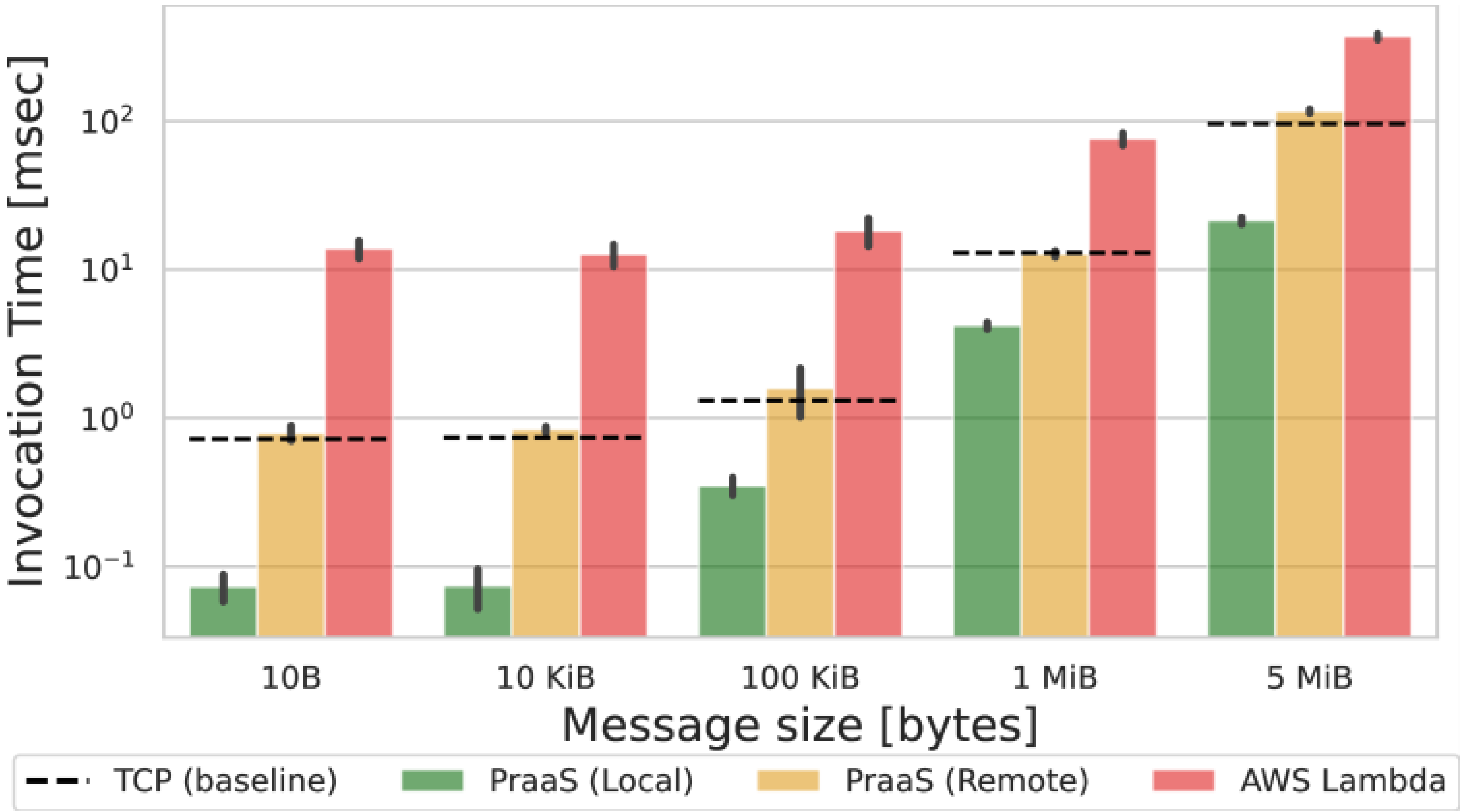


“Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes”, paper preprint.

Serverless Process on Fargate vs AWS Lambda

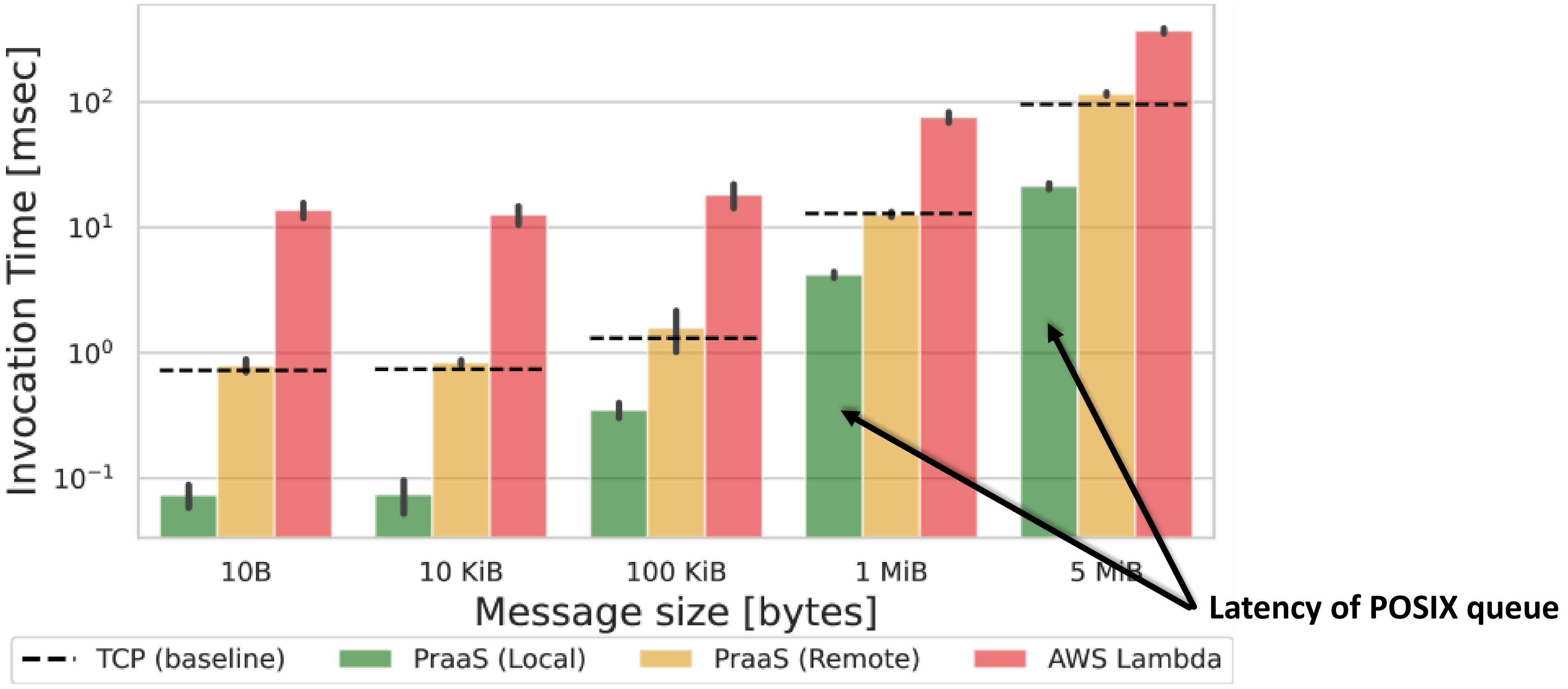
"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

Serverless Process on Fargate vs AWS Lambda



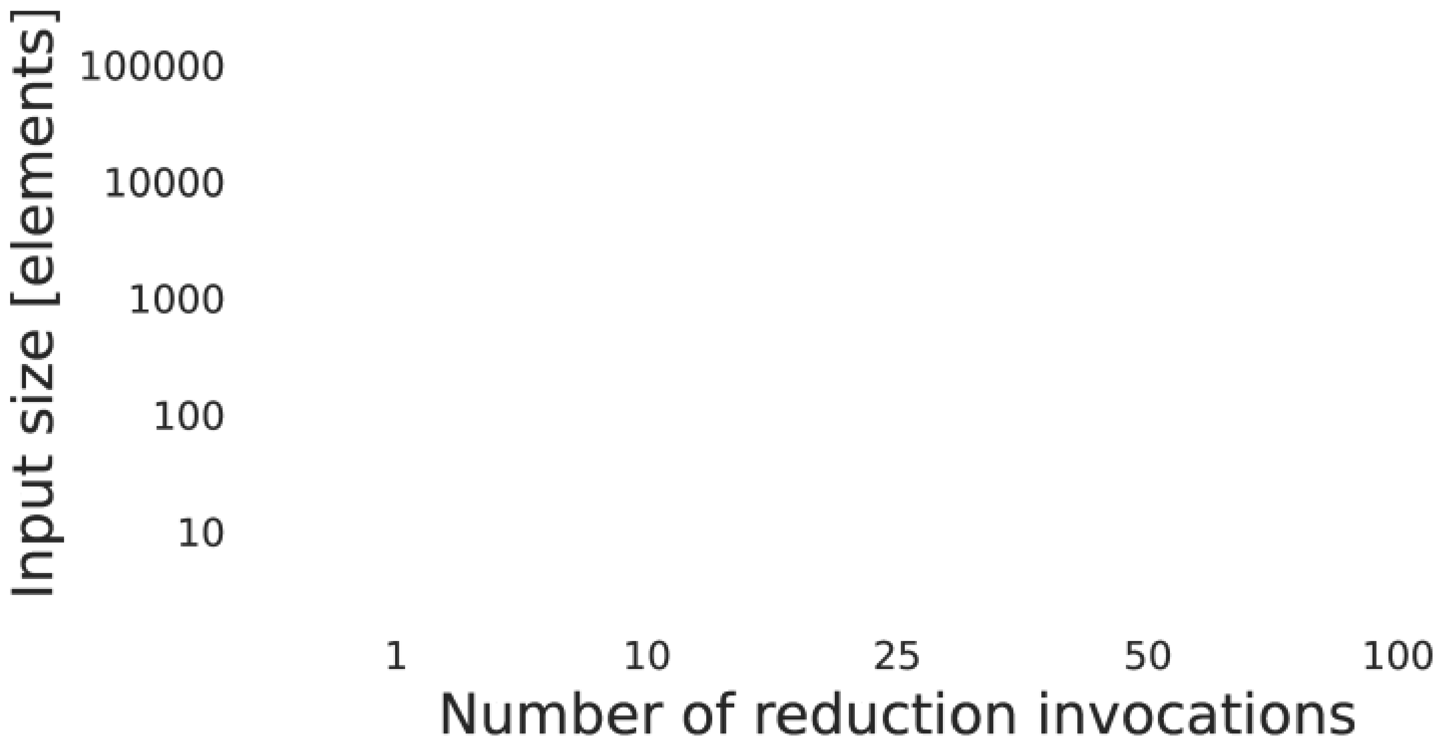
"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

Serverless Process on Fargate vs AWS Lambda



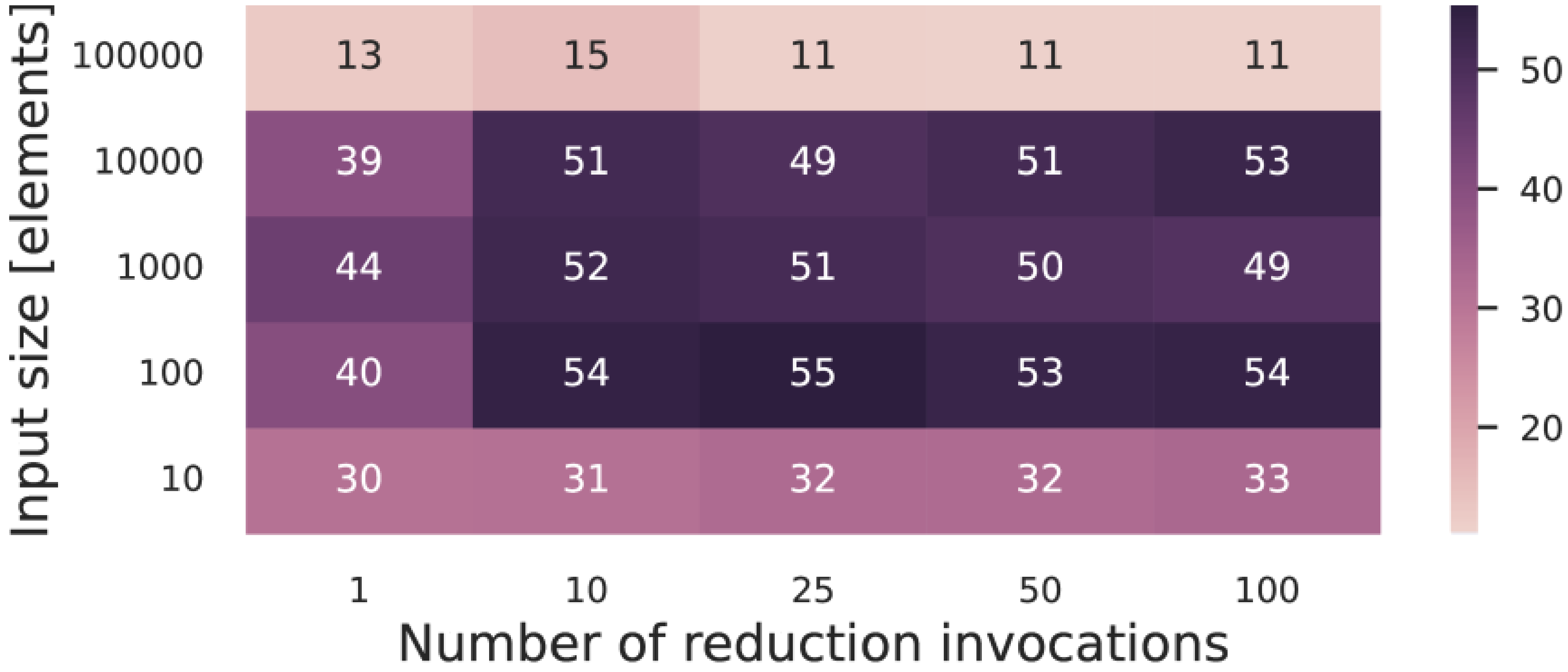
"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

Reduction Benchmark: Process State vs S3



“Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes”, paper preprint.

Reduction Benchmark: Process State vs S3



"Process-as-a-Service: FaaSt Stateful Computing with Optimized Data Planes", paper preprint.

Serverless Solutions for HPC

Serverless Solutions for HPC

spcl/serverless-benchmarks

Serverless Solutions for HPC

spcl/serverless-benchmarks

spcl/fmi

Serverless Solutions for HPC

spcl/serverless-benchmarks

spcl/fmi

spcl/rFaaS

Serverless Solutions for HPC

spcl/serverless-benchmarks

spcl/fmi

spcl/rFaaS

spcl/PraaS

Serverless challenges in HPC

Serverless challenges in HPC

Poor vertical integration

Serverless challenges in HPC

Poor vertical integration

Expensive computing

Serverless challenges in HPC

Poor vertical integration

Expensive computing

Lack of heterogeneity

Serverless challenges in HPC

Poor vertical integration

Expensive computing

Lack of heterogeneity

Restricted environments

Serverless challenges in HPC

Poor vertical integration

Expensive computing

How to integrate functions?

Lack of heterogeneity

Restricted environments

Conclusions

More of SPCL's research:

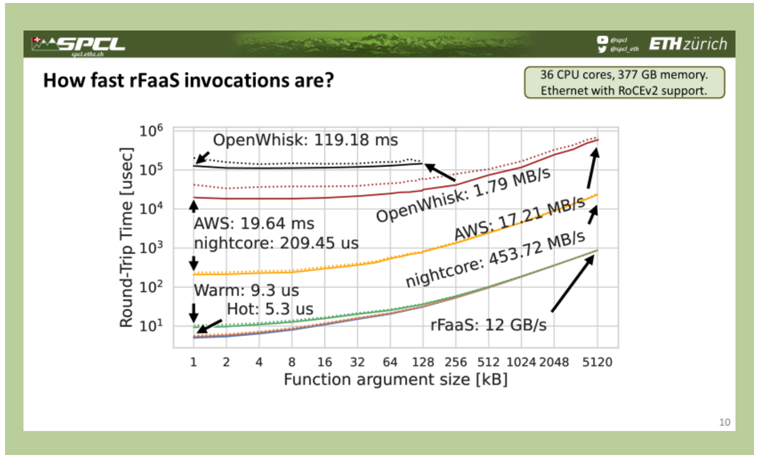
youtube.com/@spcl — 150+ Talks

twitter.com/spcl_eth — 1.2K+ Followers

github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

Conclusions

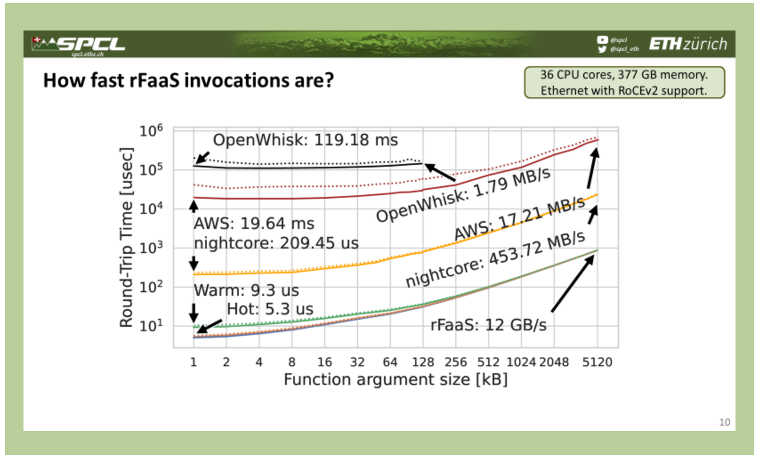


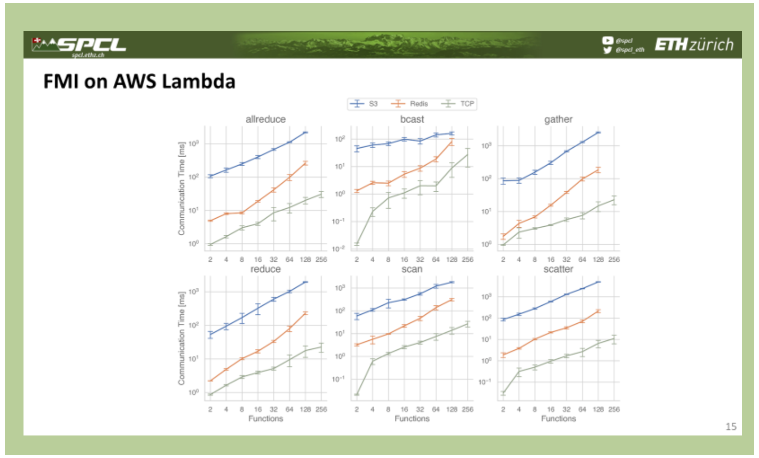
More of SPCL's research:

- youtube.com/@spcl — 150+ Talks
- twitter.com/spcl_eth — 1.2K+ Followers
- github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

Conclusions





More of SPCL's research:

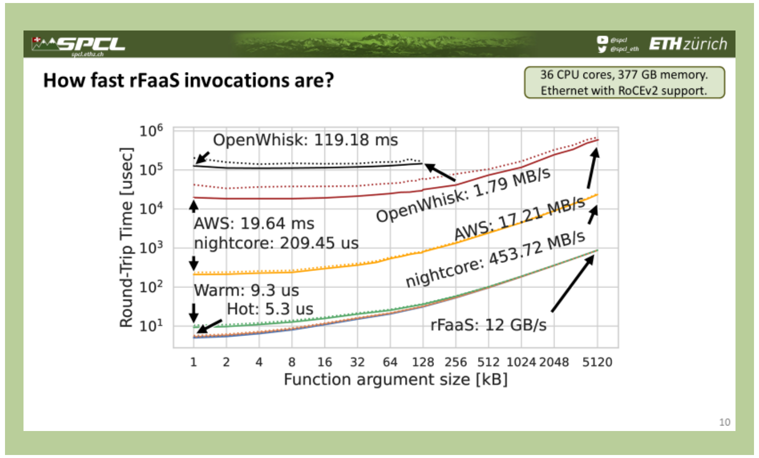
youtube.com/@spcl — 150+ Talks

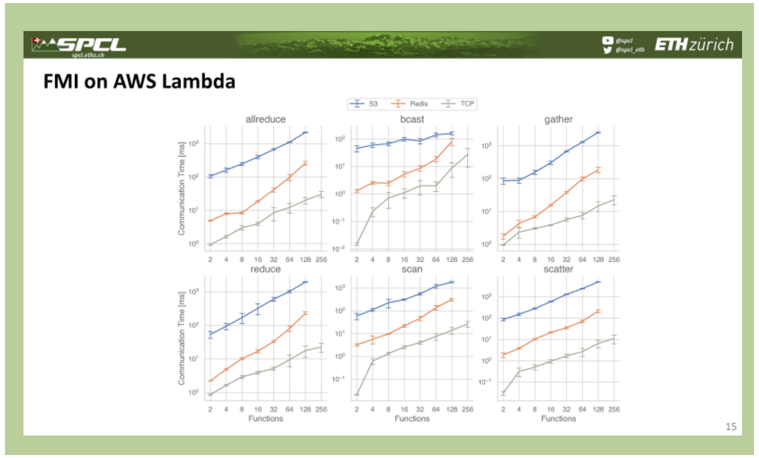
twitter.com/spcl_eth — 1.2K+ Followers

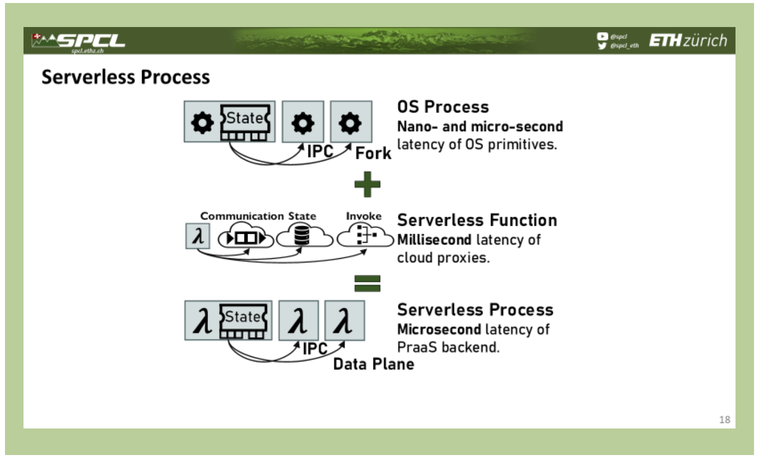
github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

Conclusions







More of SPCL's research:

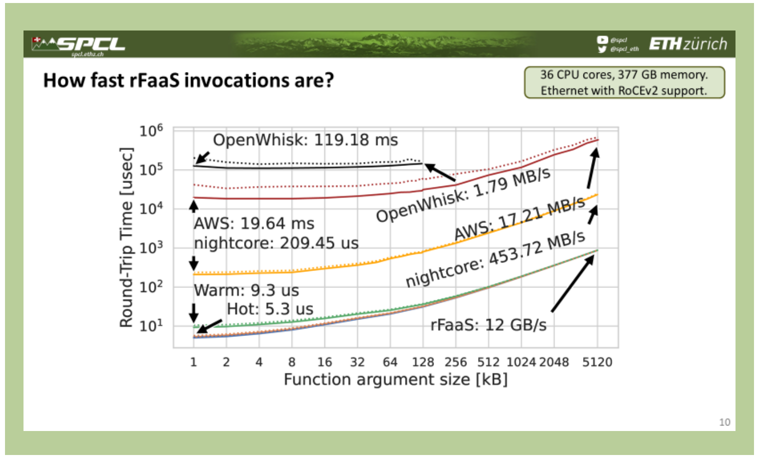
youtube.com/@spcl — 150+ Talks

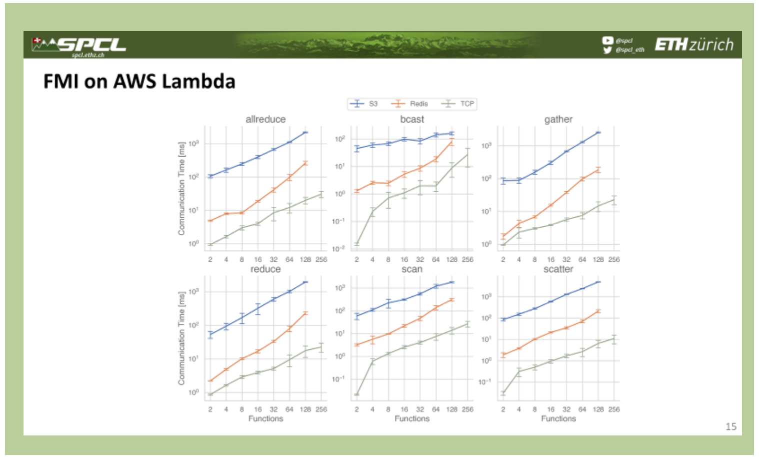
twitter.com/spcl_eth — 1.2K+ Followers

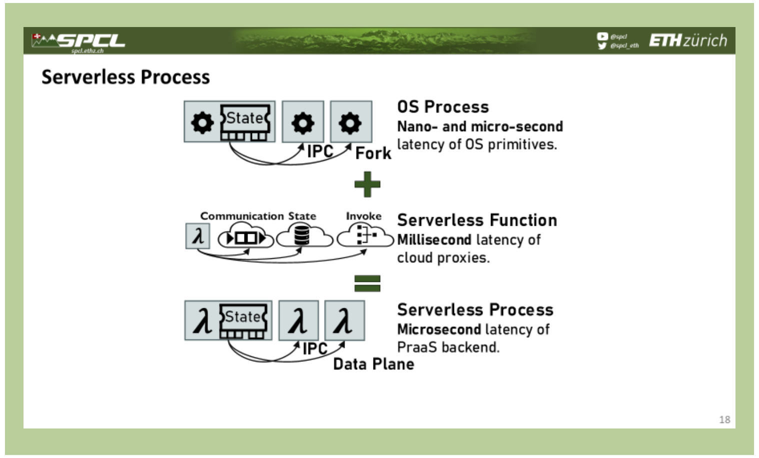
github.com/spcl — 2K+ Stars

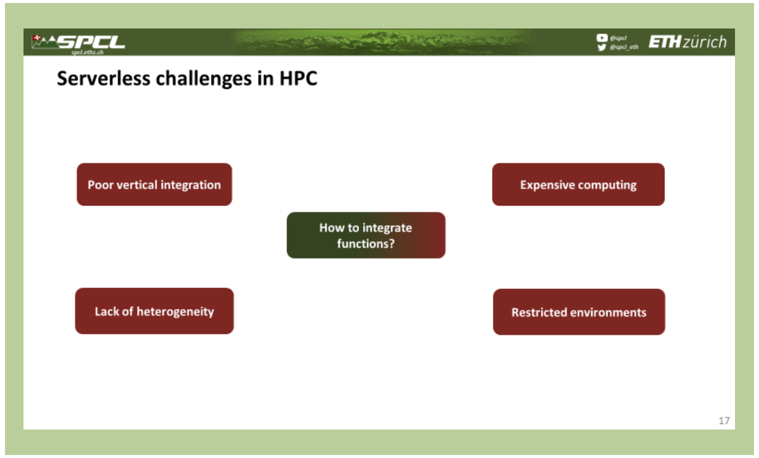
… or spcl.ethz.ch

Conclusions









More of SPCL's research:

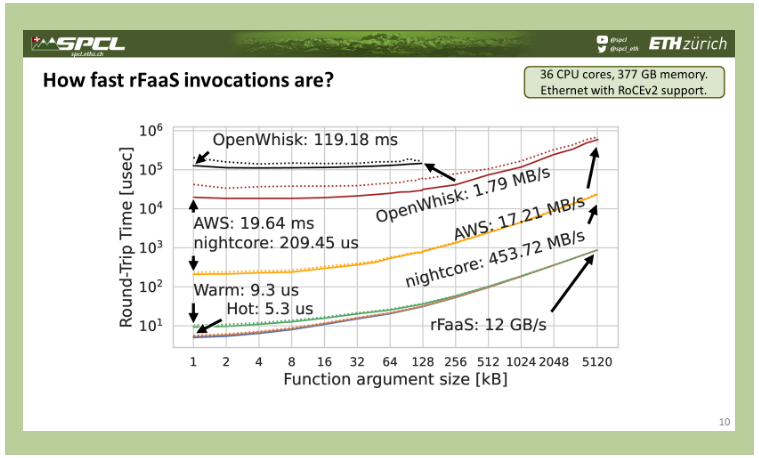
youtube.com/@spcl — 150+ Talks

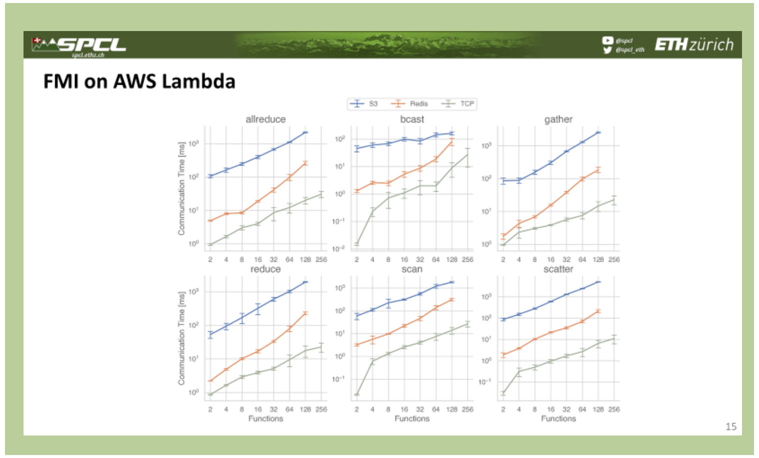
twitter.com/spcl_eth — 1.2K+ Followers

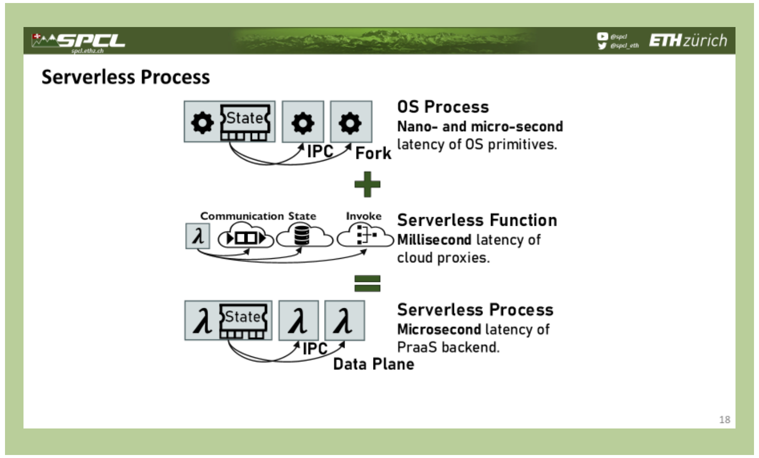
github.com/spcl — 2K+ Stars

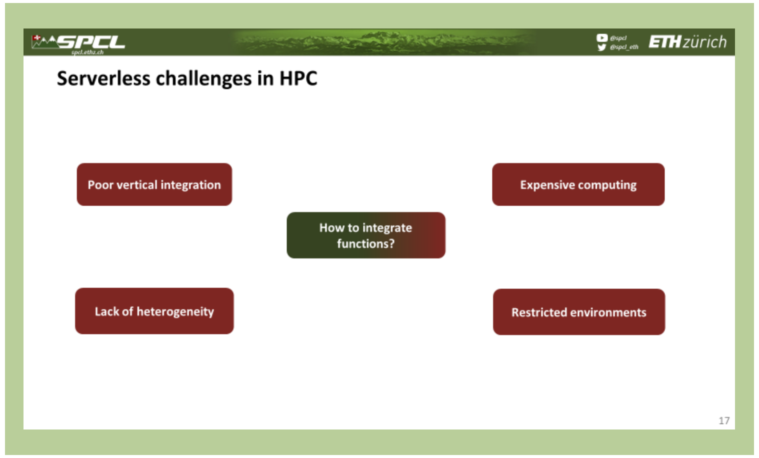
… or spcl.ethz.ch

Conclusions









More of SPCL's research:

youtube.com/@spcl — 150+ Talks

twitter.com/spcl_eth — 1.2K+ Followers

github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

Poster. **Personal website.**